DYNAMIC ROUTE REPLANNING AND RETASKING
OF UNMANNED AERIAL RECONNAISSANCE VEHICLES

THESIS

David E. R. Pritchard, Captain, USAF

AFIT/GAE/ENY/00M-10

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/00M-10

AFIT/GAE/ENY/00M-10

DYNAMIC ROUTE REPLANNING AND RETASKING OF UNMANNED AERIAL RECONNAISSANCE VEHICLES

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

David E. R. Pritchard, B.S.

Captain, USAF

March 2000

AFIT/GAE/ENY/00M-10

DYNAMIC ROUTE REPLANNING AND RETASKING
OF UNMANNED AERIAL RECONNAISSANCE VEHICLES

David E. R. Pritchard, B.S.

Captain, USAF

Approved:

Curtis H. Spenny (Chairman)	09 Mar 00
	date
Jennie J. Gallimore (Member)	06 Mar 00
	date
Steven G. Tragesser (Member)	03 MAR 00
	date

Acknowledgements

I'd like to thank my advisor, Dr. Curtis Spenny, for his flexibility, open-mindedness, and continuously open door during our work together. His guidance and thoughtful feedback were of tremendous value. The Global Hawk System Program Office was instrumental in providing up-to-date specifications, design information, and imagery on the current state of the art in High Altitude Reconnaissance UAVs. Drs. Rubin Johnson and Michael Roach of Operations Research Concepts Applied (ORCA) provided valuable technical insight into many of the desirable features of the coming generation of mission planners, as well as evaluation copies of their fine OPUS mission planning system. I thank my contacts in the Air Force, academia, and industry for all the interviews, phone calls, and e-mail letters.

My utmost gratitude goes to the love of my life, my wife Heidi. She has made the enriching AFIT experience possible, allowing me to accomplish *what* needed to be done *when* it needed to be done without complaint and generously taking up the slack when required to keep our household running. My faith that our Creator has great plans for us all (Jer29:11) and turns even our biggest mistakes into great things has carried me through the occasional strife of the rigorous AFIT program. While technical skills are useful and fun to practice, it's the life lessons from grad school about managing time, taking initiative, being self-directed and outgoing, and relating to others that I find most valuable.

Dave Pritchard

Table of Contents

List of Figures

List of Tables

List of Acronyms

ACTD	Advance Concept Technology Demonstration
AFIT	Air Force Institute of Technology
AFMC	Air Force Materiel Command
AFMSS	Air Force Mission Support System
AFRL	Air Force Research Laboratories
AIAA	American Institute of Aeronautics and Astronautics
ASC	Aeronautical Systems Center, AFMC
BLOS	Beyond Line of Sight
C2	Command and Control
CAP	Combat Air Patrol
CINC	Commander in Chief
CONOPS	Concept of Operations
DARO	Defense Airborne Reconnaissance Office
DARPA	Defense Advanced Research Projects Agency
IFRS	In-Flight Replanning System
ECEF	Earth-Centered, Earth-Fixed
ECM	Electronic CounterMeasures
ENU	East-North-Up
EO	Electro-Optical
FEMA	Federal Emergency Management Agency
FOR	Field Of Regard
GIQE	General Image Quality Equation
GPS	Global Positioning System
GUI	Graphical User Interface
HAE	High Altitude Endurance
IFMPS	InFlight Mission Planning System

IFRS	In-Flight Replanning System
IR	Infrared
JMPS	Joint Mission Planning System
JSF	Joint Strike Fighter
LO	Low Observable
LOS	Line of Sight
MCE	Mission Control Element
NIIRS	National Imagery Interpretation Rating Scale
NIIRSE	National Imagery Interpretation Rating Scale Estimate
NIMA	National Imagery and Mapping Agency
NM	Nautical Mile
NRO	National Reconnaissance Office
OPUS	ORCA Planning and Utility System
ORCA	Operations Research Concepts Applied Corporation
PC	Personal Computer
PFPS	Portable Flight Planning Software
SA	Situational Awareness
SAM	Surface-to-Air Missile
SAR	Synthetic Aperture Radar
SATCOM	Satellite Communications
SEAD	Suppression of Enemy Air Defenses
SPO	System Program Office
TAMPS	Tactical Aircraft Mission Planning System
TES	Test and Evaluation Squadron
TOT	Time On Target
USACOM	US Atlantic Command
UAV	Unmanned Aerial Vehicle
WGS	World Geodetic System
WPAFB	Wright-Patterson Air Force Base

Abstract

Today's high altitude endurance (HAE) reconnaissance unmanned aerial vehicles (UAVs) are extremely complex and capable systems. They are only as good as the quality of their implementation, however. Mission planning is rapidly increasing in complexity to accommodate the requirements of increasing aircraft and information control capabilities. Effective mission planning is the key to effective use of our airborne reconnaissance assets. Global Hawk, the current state-of-the-art in HAE unmanned reconnaissance aircraft systems, demands extremely intensive and detailed mission planning. The mission plan must accommodate a range of possible emergencies and other unplanned in-flight events, like pop-up threats or a critical aircraft system failure. Current *in-flight mission replanning* systems do not have sufficient capability for operators to effectively handle the full range of surprises commonly encountered in flight operations. Automation is commonly employed to reduce this high workload on human operators. This research proposes that a variety of common operational situations in HAE UAV reconnaissance necessitate more direct human involvement in the aircraft control process than is currently acknowledged or allowed. A state of the art mission planning software package, OPUS, is used to demonstrate the current capability of conventional mission planning systems. This current capability is extrapolated to depict the near future capability of highly automated HAE reconnaissance UAV in-flight mission replanning. Scenarios are presented in which current capabilities of in-flight replanning fall short. An improved notional PC-based mission planning system, the In-Flight Replanning System (IFRS), is then developed and presented. The same problematic scenarios are revisited with the IFRS, demonstrating improved replanning results.

DYNAMIC ROUTE REPLANNING AND RETASKING OF UNMANNED AERIAL RECONNAISSANCE VEHICLES

1 Introduction

"The Right Image, to the Right User, at the Right Time, at the Right Rate"

- Global Hawk ACTD Vision

High-Altitude Reconnaissance has been the ace in the hole for US foreign policy makers since the use of the U-2 to dispel the Russian vs. US 'bomber gap' fear during the 1950s and to uncover the presence of Soviet missiles in Cuba in 1962. We have come to depend on reconnaissance information heavily. In this area, unmanned aerial vehicles (UAVs) offer many advantages over image collection via manned aircraft. High altitude reconnaissance is aptly suited to a UAV's long endurance and the absence of risk of human life. In addition to strategic planning, today's airborne reconnaissance platforms are capable of providing vast land area coverage in near real-time; these image resources have thus become a necessary tool to tactical units as well. They are capable of providing so much real-time image data that they may actually overload human capacity to digest it. This 'information overload' is one example of how a vast capability may be limited by the quality of its implementation.

The implementation quality of an HAE reconnaissance asset is determined by the mission plan. The **mission plan** is the methodology for executing all aspects of an aircraft mission from engine start to shut down. For our purposes, an HAE reconnaissance UAV mission plan will refer to the complete set of instructions governing the in-flight operation of a single sortie of a single aircraft. It is usually developed (sometimes days or weeks) prior to beginning the mission, and may or may not be changed during the mission. Mission planning is rapidly becoming more complex to accommodate the needs of increasing aircraft and information control capabilities. An active current research area, effective mission planning is the key to effective use of our airborne reconnaissance assets.

Global Hawk is the current state-of-the-art in high-altitude, unmanned reconnaissance aircraft systems. Complexity in mission planning is the rule for complex UAV systems like Global Hawk. Missions must be meticulously planned in exquisite detail. The mission plan must accommodate a range of possible emergencies and other unplanned in-flight events, like pop-up threats or a critical aircraft system failure. Current in-flight mission replanning systems do not have sufficient capability for operators to effectively handle the full range of surprises commonly encountered in flight operations. Automation is commonly employed to reduce this high workload on human operators. For example, route planning software employing a heuristic search algorithm is commonly used to solve complex routing problems. But why not automate humans out of the control loop entirely? Victor Riley puts it candidly, "… as long as we feel a need to be able to blame someone when things go wrong, we will always want a human operator in charge" [15]. Generally speaking, *an appropriate level* of automation is good and can effectively deal with many of the complex tasks of mission planning and execution.

1.1 Purpose of Thesis

This research effort proposes that a variety of common operational situations necessitate more direct human supervision and involvement in the aircraft control process than is currently available. No capability is currently employed in HAE reconnaissance UAVs to replan segments of a mission in progress quickly and effectively when new mission objectives are identified [20]. The problem is exacerbated in situations requiring immediate action. One such situation is the incorporation into the mission plan of a **pop-up priority target**: a newly identified target of utmost urgency for image collection [16]. Scenarios are considered herein where new requirements for target imaging or threat avoidance mandate a revised mission plan within minutes of the current position [3].

These scenarios exceed the current capabilities of 'on-the-fly' route replanning. For example, if a new collection tasking comes down (typically from high level, e.g. CINC USACOM) with high priority during a mission, the operators will be required to accommodate this new tasking in a revised plan. Vastly different requirements will exist for the tool used for this mission replanning, depending on how close the target is to the UAV, how far the target is off the planned flight path, whether slack time is available in the existing mission plan, what image quality is required, etc.

The primary emphasis here is in the demanding case where the new image collection requirement is of very high importance, of variable required image quality, within 5 minutes flight time or so, and little slack time is available in the mission for slipping the rest of the route. A basic, demonstration-level route replanner is developed that lets the operator manually reconfigure a segment of the route, to be uploaded back into the master mission plan after revision. The operator draws on his or her expertise and weighs expected image quality versus survivability and the time required for the route. Many data items like target location, priority and image quality requirements can be automatically fed into the tool's database; simply adding targets or threats to a map is easily accomplished today. The contention here is that subjective judgments like weighing the relative importance of other mission requirements or changing survivability acceptance levels are handled much more effectively by situationally aware human operators. The challenge is to find a man-machine interface that is simple yet conveys the information necessary to make an informed tactical decision. Chapter 2 develops a synopsis of current mission planning practices and paradigms. Chapter 3 presents the In-Flight Replanning System (IFRS). An acronym list is provided in the prefatory material for the convenience of the reader, and Matlab code for the IFRS is detailed in Appendix A: IFRS Matlab Code.

2 Current Practices in Mission Planning

HAE UAV reconnaissance is very much still an infant technology. An operational platform does not currently exist. The Global Hawk Advanced Concept Technology Demonstration (ACTD) is the state of the art in the field. The purpose of an ACTD is to do exactly that: demonstrate and prove the viability of a superior new concept. The promise of on demand, near real time, multispectral high resolution surface imaging is alluring. Unfortunately, the process of developing such an advanced capability is daunting and fraught with difficult technical challenges. Director of Architecture and Integration, Defense Airborne Reconnaissance Office (DARO) Colonel Michael S. Francis stated in 1998,

> Despite some isolated successes, a highly publicized record of failure, replete with recent spectacular crashes, has led to the general perception that these systems, as a group, are relatively unreliable and technologically immature. While there are reasons to be optimistic today (the newest UAVs represent significant departures from earlier generations in both capability and technology), the critics' skepticism will be muted only after UAVs have proved themselves by establishing a successful test and operational track record. [7]

Today's tight budgets require tight acquisitions schedules and a low tolerance for failure, which further complicates the process. Many organizations hold hostile the encroachment of UAVs into the manned aircraft community. UAVs operating and sharing airspace with manned aircraft is a hot button issue. Though infrequent, mishaps do occur. It cannot be expected otherwise if the state of technology is to advance. Still, one midair collision or other significant accident resulting in the loss of human life would be a severe set back. It is for this reason that solution of many of the complex issues surrounding HAE UAV control are made even more crucial; the requirement for absolute rigor in establishing sometimes ridiculously wide margins of safety frequently hampers the technology development process.

2.1 The Global Hawk HAE Reconnaissance System

Figure 1 Global Hawk Air Vehicle

Global Hawk (Figure 1) is the air vehicle component of the HAE UAV ACTD. The program is executed by the Defense Advanced Research Projects Agency (DARPA) for the Defense Airborne Reconnaissance Office (DARO). The Global Hawk System Program Office (SPO) is located at Wright Patterson AFB, and is the sponsor of this research. The ACTD is currently in Phase II development and flight/payload testing [10].

The HAE UAV ACTD is “aimed at developing and demonstrating long dwell, high altitude reconnaissance, surveillance, and target acquisition” [10]. The Global Hawk air vehicle’s projected mission endurance is 40+ hours while achieving altitudes in excess of 65,000 ft and nominally operating at about 350 knots true airspeed. Global Hawk is equipped with Electro-Optical (EO), Infrared (IR), and Synthetic Aperture Radar

(SAR) sensors. The remarkable capability of these sensors to capture detail at extreme flight altitudes is demonstrated in Figure 2.

Figure 2 Global Hawk IR Image Sample: China Lake, CA from 51,000 ft. [2]

Control of the air vehicle is maintained from Mission Control Element (MCE) ground stations, where a team comprised of mission planning, command and control (C2), image quality control, communications management, and mission commander positions operates the system. There is no conventional 'stick and rudder' pilot station for the vehicle; flight paths are defined by a series of waypoints designated by the mission plan. Sensor data transmission and C2 with the air vehicle are maintained via wideband UHF line of sight (LOS) or SATCOM beyond line of sight (BLOS). Additional system specifications and general information about the program may be found in the HAE UAV Joint Employment Concept of Operations [10].

Global Hawk missions are planned with four major sub plans: the route plan, the collection plan, the communications plan, and the dissemination plan. The route plan contains all navigational information about the particular route the air vehicle will be flying. The collection plan is used to specify all imaging requirements and instructions for operating the sensors. Communications frequencies, satellite availability, and line of sight link locations, and plans for switching between all of them are contained in the communications plan. The dissemination plan dictates to whom the required images will be sent to and by what means. The four mission sub plans are defined and in place prior to mission start; this is the mission planning process. Once airborne, modification of the mission plan constitutes in-flight replanning, or dynamic replanning. In-flight replanning capabilities are currently quite limited and are still under development [10].

To reiterate, the nature of the ACTD is to accomplish specific goals in the demonstration of enabling technologies for HAE reconnaissance. It is not meant to be an operational production system, but to facilitate risk reduction and lesson the acquisition costs associated with such a major leap in UAV reconnaissance technology [10].

2.2 HAE Mission Planning: The Current Paradigm

UAVs like Global Hawk face less direct control issues on the part of their operators than do manned aircraft. Pilots of conventional aircraft are continuously answering questions such as, “Should I turn left now, at what bank angle, and how do I need to move the stick to achieve that flight condition?" Global Hawk is of the emerging new generation of UAVs which fly using only supervisory control by its operators. On-board flight control software handles the details of maintaining bank angle, heading, airspeed, etc. Since the details of flying the aircraft are left to the flight control software, the human operators may concern themselves with broader questions. The MCE crew is concerned with the big picture: "Where are we now, where do we need to be, when do we need to be there, and what do we need to do along the way?" When they become available, mature in-flight mission replanning systems will better help operators facilitate answers to these questions.

Long mission durations coupled with dynamic intelligence environments result in a dynamic image collection requirement. It can be expected that over the course of a 40-hour UAV mission image collection requirements will frequently change after the mission plan has been uploaded to the aircraft. Much work is currently being done to develop automated mission control software for UAVs that reduces operator workload and allows usually one person to control the route tasking of the aircraft.

Many aspects of the Global Hawk system are still undergoing significant development. Currently, development of the mission planning software is in this category. Planning a Global Hawk mission from the ground up takes several weeks as of this writing. Much of this time is taken up doing tasks not representative of an operational environment, however. For example, a significant headache is currently making contingency plans to divert to the precious few alternate landing sites specially equipped to land a Global Hawk. This is necessary should an in flight equipment failure or other emergency dictate the premature termination of a mission, but this extra mission planning tedium will be reduced when the system becomes operational and many more landing sites are available [20].

Keeping human operators out of the direct flight control loop and instead playing only a supervisory role offloads the bulk of complexity to automation. This is highly effective for many, and perhaps most, in-flight tasks. The Global Hawk ACTD has adopted this **supervisory control** approach. Human operators supervise the autonomous execution of the mission plan rather than directly manipulating control surfaces.

2.2.1 The Human Interface

In general, it is common practice to implement automation to counter the increasing complexity of today's systems; be it UAVs, computer-aided manufacturing processes, or Denver International Airport's multimillion-dollar luggage ground routing system [22]. However, humans are ultimately responsible for the safe and effective operation of these systems regardless of the level of automation. The human interface is our 'window' to the activities of automated processes and the means by which we supervise and control them. Human interface research has shown that automation (vs. manual control) carries its own inherent complexity, which may or may not be an improvement. In fact, an ineffective human interface may not relieve complexity, but simply realign it. This is an overriding concern in human interface design for mission planning systems [22]. Figure 3 shows the In-Flight Mission Planning System (IFMPS) Pilot Vehicle Interface (PVI) described in section 2.3.1. The IFMPS PVI is a good example of a well thought out user interface, conveying an appropriate level of information to the user for the tasks for which it was designed.

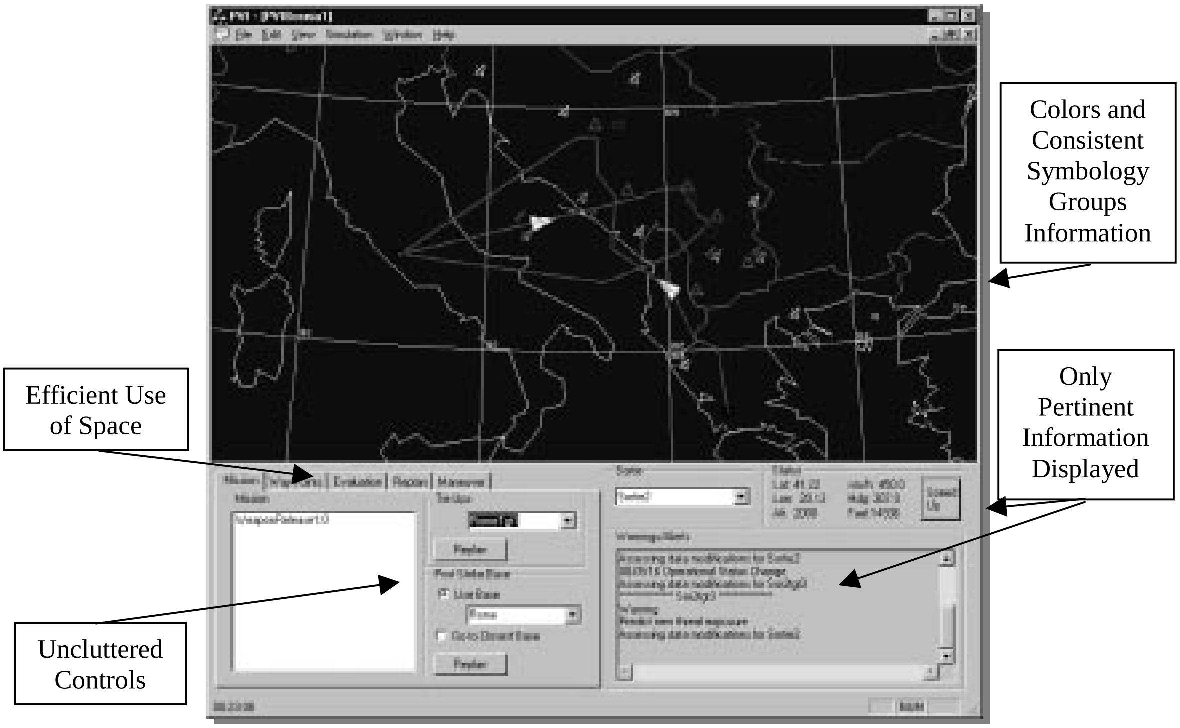

Figure 3 ORCA's IFMPS PVI

2.3 Mission Planning vs. In-Flight Mission Replanning

UAVs like Global Hawk that take a supervisory tact to flight control and mission execution rely heavily on complex software systems. In the case of Global Hawk, the development of this software by Marconi is, in fact, currently one of the driving efforts in the advancement of the Global Hawk ACDT. One subsystem of this software handles the directing of the air vehicle along its prescribed flight path, accomplishing its objectives along the way. The user interface of this route planning subsystem must facilitate many of the same tasks as those done in mission planning before takeoff. Thus, a duality exists between the requirements of mission planning and in-flight mission replanning systems. **In-flight mission replanning**, also called **dynamic replanning,** is the amendment of an existing mission plan during execution. Mission planning is a complex process, and it would seem intuitively obvious that replanning in-flight should be much more difficult. This is in fact the case. Because of the time-critical nature of mission execution and the fact that mission changes cascade down to affect future route sections, in-flight mission replanning is very challenging to facilitate and slow in development.

Mission planning software for UAVs like Global Hawk performs many of the same tasks as conventional mission planning software for conventional aircraft. In fact, whether a mission planning system is being used to preplan a mission or to control a UAV mission in flight becomes virtually transparent to the user interface. In other words, a mission planning system could be made to operate much the same way as the front end to the actual flight control software for a UAV. Whether a mission planning system is being used to plan a mission for an F-16 sortie next Wednesday or for a Global Hawk UAV currently in flight is transparent to the function of many mission planning system components. Therefore, it is conceptually simple to extend the current capabilities of today's mission planning systems to portray future capabilities of in-flight mission replanning. OPUS, one of today's most advanced mission planners, is described in the next section. It is used to represent in-flight mission replanning capabilities of the near future.

Let us draw one further distinction between dynamic replanning vs. dynamic retasking: dynamic replanning entails making quick changes to the existing mission plan, in whole or in part. This includes changing the flight plan, collection plan, communications plan, or dissemination plan. The current scene being taken may be aborted, and future collects may be changed or deleted. **Dynamic retasking** implies the ability to quickly change plans for image collection tasking without changing routes or other aspects of the overall mission plan [10].

The current capability for in-flight replanning is very limited. Significant technical advancement is still being made in other aircraft systems. The current state of maturity of UAV mission control environments is focused on the more basic elements of mission execution; the primary one being a safe, full stop on the runway. Current HAE UAV operations are in the technology *demonstration* phase. Functional details and specifications of operational capabilities are now being notionally developed.

2.3.1 Existing In-Flight Replanning Research and Demonstration Software

Many organizations are currently conducting research in the in-flight mission planning area. Among them are the Crew System Interface Division, Human Effectiveness Directorate, AFRL (AFRL/HECI) at Wright-Patterson AFB, and OR Concepts Applied (ORCA) Corporation of Whittier, CA. A Phase II program entitled, "Low Observable Inflight Replanning" was undertaken by ORCA. Managed by HECI, the program resulted in significant findings (presented in The IFMPS Final Report [13]) and the Inflight Mission Planning System (IFMPS). The IFMPS is primarily geared towards the dynamic replanning of operational, low-observable (LO) aircraft (B-2, F-117, etc.) mission plans. They found that basically three situations result in the need to replan in-flight: a change in threat environment (e.g., pop-up SAM); a change of mission requirements (e.g., new targets); weather hazards. "New information is the motivation," states one viewgraph bullet about what drives in-flight replanning. Much of the concept of the IFMPS can be extended to UAV mission planning [17].

2.4 Primary Mission Planning Considerations

For reconnaissance missions, the most important product is the image dataset. Image quality may vary considerably depending on many factors, and is often the driving issue when planning a mission. To quantify image quality, the National Imagery and Mapping Agency (NIMA) developed National Imagery Interpretation Rating Scales (NIIRS) for various sensor types. On a scale of zero to 9, NIIRS ratings state the degree of detail interpretable from the image in final form (after post-processing and when viewed in a controlled environment with calibrated monitors or hard copy). NIIRS ratings may be predicted using the appropriate General Image Quality Equation (GIQE) (see section 3.4.1) [15].

In hostile airspace, image collection will occur at the cost of exposure of the air vehicle to threats. These threats may be air to air in the form of hostile aircraft, or surface to air in the form of missiles. A threat exposure metric is commonly computed based on exposure to hostile radars, with increasing penalty for search, tracking, or fire control radars, respectively. Threat models are defined for each known threat type expected to be encountered.

Time is a primary consideration for mission planning for multiple reasons. First, given a bounded mission duration (i.e. in the absence of air to air refueling), less time spent fulfilling one mission objective equates to more time available for adding more objectives later. Second, some objectives may be constrained by a particular time requirement. These “time-on-target” (TOT) specifications require the objective to be reached at a certain time. Examples would be the surveillance of a scheduled event, joining a flight of other aircraft, or an air-to-air refueling appointment.

Manned aircraft missions are duration limited by a human’s capability to remain continuously mission-capable inside the aircraft. UAVs do not share this limitation. Operators may be rotated on shifts to provide continuously fresh human control capability. How to maintain situationally-aware operators over the course of mission durations exceeding 40 hours and through several crew changes becomes a significant challenge, however. It is conjectured here that situational awareness (SA) is maintained most effectively when operators are deeply involved in 'big picture' decisions about the route planning process. This is also one of the primary considerations in mission replanning: how to maintain operator SA, and at what level [7].

2.5 Additional Mission Planning Considerations

By definition, in-flight replanning must include the altering of a pre-planned course. Flexibility must be available in the overall mission plan to allow the aircraft to be in previously unexpected places at unexpected times. Deconfliction of aircraft paths sharing the same theater airspace is therefore paramount. Because of the high-altitude nature of the Global Hawk platform, conflicting flight paths with other aircraft during cruise are not probable and easily avoided. Any other aircraft at 50,000 ft or higher would likely be other Global Hawk UAVs, all of which would be under control of a common MCE, or at least multiple MCEs linked via their common network. At any time during ascent or descent, however, deconfliction of flight paths is a primary concern.

Other items that must be considered when planning missions are alternate landing sites, which must be available should an in flight emergency arise. This is of particular concern for current Global Hawk ACTD operations (i.e. *test* operations) in the face of high-visibility and few spares. Arranging alternate landing sites currently accounts for a significant percentage of time spent on initial planning of missions.

2.6 What Information Should be Presented to the Operator During Flight?

It depends on the time available for replanning. Available reaction time for replanning tasks generally falls into three timeframes: hours, minutes, or seconds [17].

- <u>Hours</u> available for replanning: Scenario Example: A change of landing base is required due to weather, with plenty of time to make changes [17].

Plenty of time is available to reconfigure the necessary route parameters to account for a new landing base. All aspects of the mission may be considered during the replan, so the operator may require great detail from a complex user interface. The route could be automatically re-optimized by a search algorithm.

- Minutes available for replanning: Scenario Example: Change of target. A high priority new target is designated only several minutes in advance [17].

The new target’s priority may supercede all other targets, some, or none of them. Sufficient time in the 'minutes' category is not available for optimal route regeneration by changing or loading in all new mission parameters. Only a bare minimum of detail should be presented to the operator so as to not waste attention on unnecessary or irrelevant data. Because a quick solution is paramount, acceptance of a sub-optimal route may be necessary. Additionally, the relative priority of mission objectives is not always readily quantifiable: the situation typically requires intervention by the operator in a timely manner. This makes a high level of operator SA critical. The ‘minutes’ timeframe is the focus of this research and of the IFRS.

- Seconds available for replanning: Scenario Example: A previously unknown SAM site illuminates the aircraft with fire control radar. Immediate action is required, and possibly evasive maneuvering. Human intervention may take too much time; the execution of preprogrammed maneuvers may be necessary [17].

In the 'seconds' timeframe, sufficient SA may not be immediately available for the operator to step in and assume guidance of the aircraft as effectively as a series of preprogrammed evasive maneuvers. These maneuvers may be executed automatically and the operator simply notified of the action, with the option to override. A primary strength of UAVs is the offloading of mundane operator tasks to automation. It is conjectured here that decreasing operator involvement has the consequence of reducing situational awareness. If a human were to intervene effectively in this “immediate action required” timeframe, a high degree of situational awareness is necessary if the operator is to be able to immediately take control of the aircraft and direct it away from the threat.

2.7 Current Mission Planning Systems

Several powerful mission planning software packages are available and being used in the military community today, the major ones described below. Others are in more limited use like the TLAM (Tomahawk Land Attack Missile) Planning System (TPS). Still others are simply in earlier stages of development, like the system Boeing is developing for the Joint Strike Fighter (JSF). The common denominator is that they all strive to take full advantage of the maximum capability of the aircraft system. The result is a very capable, albeit very complex system and user interface. As one would expect, the systems require significant training for operators to become proficient in their use for live operations [8].

2.7.1 The Air Force Mission Support System (AFMSS) Family

AFMSS is built around a core of UNIX based programs and modules. Plug-ins called Aircraft, Weapons, and Electronics (AWE) modules are system-specific to each aircraft being served. The Common Low Observable AutoRouter (CLOAR) package for LO aircraft like the B-2 and F-117 is an example of the many software programs in the AFMSS family. AFMSS provides Air Force users with a tool that speeds such aircraft specific calculations as fuel requirements, etc. and eliminates many of the mundane details of mission planning [8].

2.7.2 Portable Flight Planning Software (PFPS)

PFPS originated as a creation of Air Force personnel to fulfill the desire/need for a PC based mission planning system. It is government owned and maintained, and has been adopted as the PC system of AFMSS. Major components include Combat Flight Planning Software (CFPS), FalconView; Combat Weapon Delivery Software (CWDS), Combat Airdrop Planning Software (CAPS), and Cartridge Loader (CL) [8].

2.7.3 Tactical Aircraft Mission Planning System (TAMPS)

TAMPS is used by the US Navy and Marine Corps for the mission planning requirements of their fixed and rotary wing aircraft. It is comparable the Air Force's AFMSS in that it has many advanced capabilities and runs on a UNIX platform. The Navy also is using PFPS as an interim PC based system until the Joint Mission Planning System (JMPS; see below) becomes available [8].

2.7.4 Joint Mission Planning System (JMPS)

JMPS is to be the replacement system for AFMSS and TAMPS. It is a joint development between the Air Force and Navy. Both AFMSS and TAMPS have suffered human interface deficiencies according to users, which JMPS will strive to correct. This multi-service mission planning system will provide commonality for improved interoperability during exercises and training regimen [8].

2.7.5 The OPUS Mission Planning System

A state of the art mission planning software package, OPUS demonstrates the current capability in today's mission planning systems. OPUS contains the ability to control virtually every aspect of many conventional aircraft missions. It could also be adapted for use to control the mission execution of UAVs. The mission planning environment can be dynamically linked to military intelligence updated databases containing target and threat information. Routes are normally defined by an autorouting algorithm, which finds an optimal path by minimizing a cost function, or weighted combination of several parameters. These cost function parameters are typically such things as distance traveled, exposure to threats, and fuel consumption. The route is constrained to fly through predefined route points, which denote targets for weapons release, reconnaissance targets, rally points to meet up with other aircraft elements, or refueling appointments. These route points along with the order in which they are visited are called a Tie-Up. Route points may or may not have Time-on-Target (TOT) constraints, further constraining the route to pass through the specified location at a specific time. OPUS's autorouter uses an A-star heuristic search algorithm for very fast computation. The autorouter is very flexible and can accommodate the needs of virtually any situation, allowing the operator to vary parameters like "tenacity", "prudence", "economy", "curiosity", "discretion", and "caution". Through these parameters, the user controls the cost function weights for the autorouting optimization problem. Thus, the user has precise control over the route solution, and can tailor the solution to the needs of the mission at hand [12].

OPUS is not currently tailored for use in highly specialized reconnaissance aircraft like Global Hawk. It does not contain sophisticated sensor models representative of advanced EO, IR, or SAR systems [12]. Neither does OPUS currently allow the variation of aircraft proximity to a target for image collection during automatic route generation, nor weighting of this proximity into the autorouting cost function [12]. In other words, OPUS was not designed for high altitude reconnaissance where standoff ranges from aircraft to target are frequently significant and variable. Standoff range may be increased intentionally depending on the mission. In the case of EO and IR sensors, image quality may be traded off for increased separation distance and therefore safety from a nearby SAM site. In the case of SAR, image collection below a given standoff range is not even possible with current systems, and image quality increases with standoff range under some circumstances. The IFRS incorporates the significant feature of *image quality prediction* vs. standoff range and aircraft geometry into the route planning process.

3 The In-Flight Replanning System (IFRS)

The IFRS software tool is developed to demonstrate how a simple man-machine interface combined with an experienced human operator can be the most effective way to solve specific mission planning problems. It is to be used in conjunction with an OPUS-like **master in-flight mission replanner.** The IFRS would be employed as a sub system of this master replanner in specific circumstances to facilitate quick replanning of a small segment of the overall master mission plan. The replanned mission segment is then returned to the master replanner for splicing into the original plan, which is then updated and reoptimized as time is available. Figure 4 graphically depicts the notional in-flight mission replanning process considered herein.

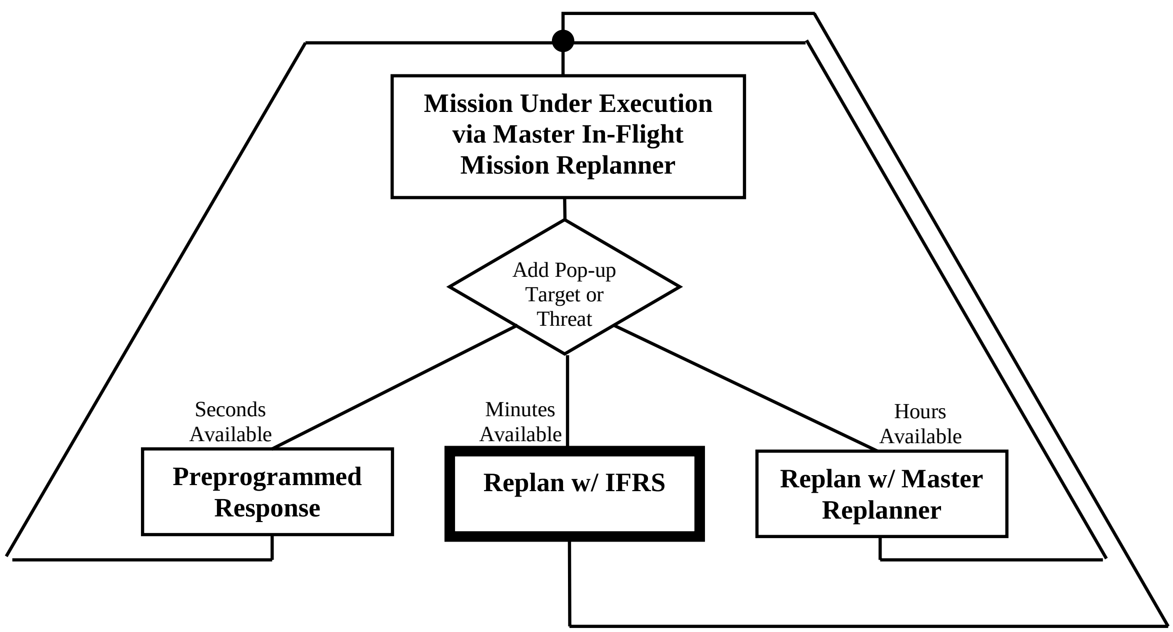

Figure 4 Overview of IFRS Employment

Several scenarios are presented in the next section, followed by IFRS application to the scenarios in section 3.5. These examples strive to make the case for IFRS mission replanning capabilities, allowing a high degree of human involvement with the added benefit of good SA maintenance. It should be stressed that these properties are not currently available with today's limited in-flight mission replanning capabilities. The dynamic environments demonstrate how operators must be able to quickly assess relevant mission data and then command the aircraft to take appropriate action.

3.1 A Poignant Scenario

A notional surveillance mission of selected targets in the United States was developed for demonstrating a rerouting response to the following scenario. The United States was chosen for the reader's ready recognition of distance scale and landmarks.

Figure 5 Levee Scenario Participants

Fictional Scenario: As part of a series of dual purpose training and flood monitoring missions, the 31st TES Global Hawk operators, in cooperation with the

Federal Emergency Disaster Agency (FEMA) (Figure 5), are conducting infrared scanning of levee integrity and flood progression throughout the Missouri River flood plain. Waters are currently 10 feet above flood stage in some areas. IR imaging missions are being conducted at dusk to take advantage of reduced sun-induced background noise levels. Temperature gradients resulting from differential solar heating of levee materials and floodwater reveal relative barrier strength characteristics. Optimum contrast and thus levee breach prediction accuracy is achieved with a side view of the levee at sunset with 90% prediction quality degradation after 1 hour. Departing from Edwards AFB, CA at 17:00 local time, RQ-4A Global Hawk will fly a regularly scheduled night surveillance mission of the Missouri River flood plain. Primary targets include a levee in danger of being breached near the town of Glasgow, MO, and another one protecting St. Louis, MO. Secondary targets include a forestry survey in Washington State and mapping a brush fire in Florida's Everglades National Park. Figure 6 depicts the flight plan route.

Figure 6 Map of Flight Plan

Several minutes before surveying the first levee, the MCE is called upon to participate in an emergency rescue operation nearby. A river tributary has diverted around barriers and washed through a town, decimating houses and cars. An unknown number of residents have been swept into the raging waters. Global Hawk operators must deviate from the current mission plan and conduct a IR area search concurrent with ground units to possibly locate and obtain a count of trapped victims in the disaster area. Coordination is through FEMA.

The town's target coordinates are uploaded to the mission plan database and Global Hawk rerouted to image the devastated town. The mission commander must decide how to prioritize the upcoming sequence of events. The images coming back from the town show many people, but it is unclear which are victims and which are searchers. Nonetheless, communication and assistance with the ground search crews continues. Should the aircraft break off from the search? The levees might not make it another 24 hours in time for another mission tomorrow night. Many lives depend on knowing where the levee danger zones are and how bad they actually are.

Throughout the crisis situation, the mission commander must continually make informed, moment to moment decisions about surveillance target priorities. It will be crucial that the mission commander have the ability to quickly direct the aircraft and sensor tasking to accommodate a dynamic environment. A simple, direct way of directing the aircraft and its sensors is crucial.

3.2 Other Examples

The previous scenario took place over friendly territory. When operating in hostile airspace, additional considerations must be made for threat avoidance and mission planning complexity increases. Minimum requirements may be placed on image quality, in addition to changing target priorities as new intelligence information arrives. The following examples illustrate the dynamics of possible operational environments.

3.2.1 Example Senarios- Replan with Master Replanner (IFRS Not Required)

- Pop-up threat appears in flight path, but nearest target priorities are such that the image must be collected; Disregard threat

- New target is added to theater in low threat area, and it can be accommodated in existing target list without breaking time constraints. Increased threat risk not a factor; Assimilate new target into route and collect at maximum image quality

3.2.2 Example Senarios- Replan with IFRS

- Example 1: Minimum Image Quality Specification and TOT Constraint
    - A new pop-up high priority target is added to theater.
    - Target is 5 minutes away.
    - New target makes 4 surrounding, preexisting targets also high priority.
    - Region protected by SAM coverage.
    - Requires NIIRS > 5 for new target; > 7 for all other targets
    - Must over fly a far future target at a prescribed time, such that not enough time is available to accommodate the new target and all preexisting targets: must skip some upcoming targets, all of which have comparable priority. Must decide which group to skip based on operator's expert knowledge of theater.
- Example 2: Threat Risk Not Quantifiable
    - A pop-up threat appears nearby.
    - Global Hawk is testing a new ECM package onboard: effectiveness against various threats is unknown/unproven, threat models in autorouting algorithm may be too conservative.
    - Targets of medium priority are being imaged now.

- Threats of this type have missed 8 of 10 shots over the last week; intelligence suggests ECM may be moderately effective against this threat type.
- The operator must weigh relative target priority and image quality requirements vs. reduced risk from threat.

3.3 The Need for an IFRS

Given that mission planning software must be very complex to control all facets of HAE UAV reconnaissance, the interface must throughput huge varieties of data. The same flexibility of user interface that allows the user to solve planning problems by choosing the most effective options becomes a liability when time critical problems are encountered. It takes time for human operators to wade through many options and choices, setting radio buttons, list boxes, pull down menus, and editable text boxes as they go. Common mission planning tasks (e.g. change the weight of threat risk vs. routing efficiency or re-order a group of targets) in current, conventional mission planning systems may take 20 or more discrete operations. This makes operators wish for an abbreviated way of accomplishing certain tasks when time is short. To restate the problem, only the most absolutely necessary choice options and information must be presented to operators when time-critical situations dictate a speedy and potentially sub-optimal solution. The phrase "quick and dirty" is sometimes appropriate to referencing solutions that only must be good enough to get the job done. Microsoft Windows CE, a bare bones operating system for personal assistants and palm size computers, is a good example of how a complex system like Windows can be stripped down and made useful for simple, quick tasks.

Global Hawk operators from the 31st TES have been interviewed who end their program status briefings with a chart that simply has the words "Mouse Clicks" in a hatched circle, like a no-smoking sign (Figure 7) [4].

Figure 7 No Mouse Clicks [4]

It takes time to change parameters and navigate through all the fields and menus of albeit extremely capable mission planning software. It is argued that operators need the ability to pare down very complex mission planning capabilities to a minimum level when speedy, on-the-fly decisions are required.

Intelligence information surrounding the mission environment can change quickly, as well as requirements for intelligence collection by the image data end-users. Simply adding new targets and threats to the mission theater during mission execution is not a problem under many circumstances. Changing objectives can often be loaded directly into databases accessed by the route-planning algorithm. These autorouters can easily optimize a new route to accommodate new targets or avoid new threats if enough advance notice is given for human operators to quantify the new situation and physically input this data. Some scenarios, however, present ‘fuzzy’ or less clearly defined target or threat properties. These are difficult to quantify and load into an autorouter's database, especially when time is short. Colonel Michael Francis from DARO has also said, "The human capability to synthesize complex forms of information and rapidly render judgment is superior to today's computer-based systems in many, if not most, circumstances" [7]. The solution is to let the human operator keep track of this information and not worry about converting it to discrete parameters for the autorouting software. This research effort is to develop a demonstration-level route replanning tool which allows the operator to quickly replan a segment of the route based on three simple route 'quality' metrics: threat risk, expected image quality from the required targets, and time available to fly the route.

3.4 Software Architecture

A full-capability, OPUS-like in-flight mission replanner is used to control all aspects of the mission during the normal mode of operations. It handles the incorporation of new targets, threats, image quality requirements, prioritization of tasks, alternate landing sites, etc. into the master plan of executing all mission objectives. The master planner has full autorouting and route quality analysis capabilities, and every option is available to the human operators to take full advantage of the Global Hawk's vast array of capabilities. Should a situation arise requiring a "quick and dirty" replanning solution, the IFRS is employed.

The IFRS functions as the Windows CE mode to the main mission planner. It is used to allow the operator to develop a modification to a selected segment of the route, usually a section within several minutes of being over flown. The selected segment is downloaded to the IFRS, modified by the user, and then uploaded back to the main mission planner to be incorporated into the master mission plan. The route following the new segment is then re-optimized with the autorouting algorithm. Considerations for keeping time-on-target requirements for future route points (one example being when fuel is exhausted) are presented to the user as part of the feedback given during the route modification process.

IFRS computes the route quality metrics NIIRS Estimate (NIIRSE) (see section 3.4.1) and simulated route survival probability. Available slack time is also computed, which may be used for extending the distance traveled in the segment being modified. A slack time limitation is present only if future TOT constraints exist. For instance, if an upcoming target must be surveiled by sundown at 18:30 and the current route allows the image to be collected at 18:00, 30 minutes of slack time is available. This 30 minutes may be used to extend the route before that target to avoid threats or collect additional images.

After running the initial calculations on the segment and displaying the results, the user has two options for modifying the route in the IFRS (Figure 12):

- **Create Route** Mode: click to define waypoints of a new route between start and end points.
- **Adjust Route** Mode: shape, or ‘tweak’, the existing route by dragging waypoints

Following definition of a new route, the user presses the **Evaluate Route** button to recompute and display the new route quality metrics. The process continues iteratively until either the route fulfills all requirements, or it is judged ‘good enough’ because time has run out. The newly replanned segment is then uploaded back to the master in-flight mission replanner for assimilation into the existing route and subsequent reoptimization.

3.4.1 National Imagery Interpretation Rating Scale Estimation (NIIRSE)

NIIRSE is a numerical prediction of NIIRS rating by the IFRS for a given imaging geometry configuration, i.e. azimuth, elevation, and slant range. In an operational situation, an intensive calculation using the GIQE would be most accurate for making NIIRS predictions. EO and IR GIQEs have a multitude of inputs and situation-specific variables [15]. Some are sensor dependent, while others depend on target materials and surroundings, atmospheric conditions, and geometry. Often, the target type and some information about its surroundings will be known, as well as pertinent sensor parameters. Atmospheric data and lighting conditions may be measured or calculated, and incorporated into the computation as well. Given the availability of this information to a sophisticated in-flight mission planning system, predictions of image quality may be made in flight with reasonable accuracy from the GIQE. The data points in 0 represent NIMA-assigned NIIRS ratings of Global Hawk IR imagery. Vertical bands show the 95% confidence intervals of prior NIIRS predictions using the GIQE. The logarithmic relationship between NIIRS degradation and (standoff) range is also apparent. Specific values have been omitted from the plot due to the sensitive nature of system specifications [2].

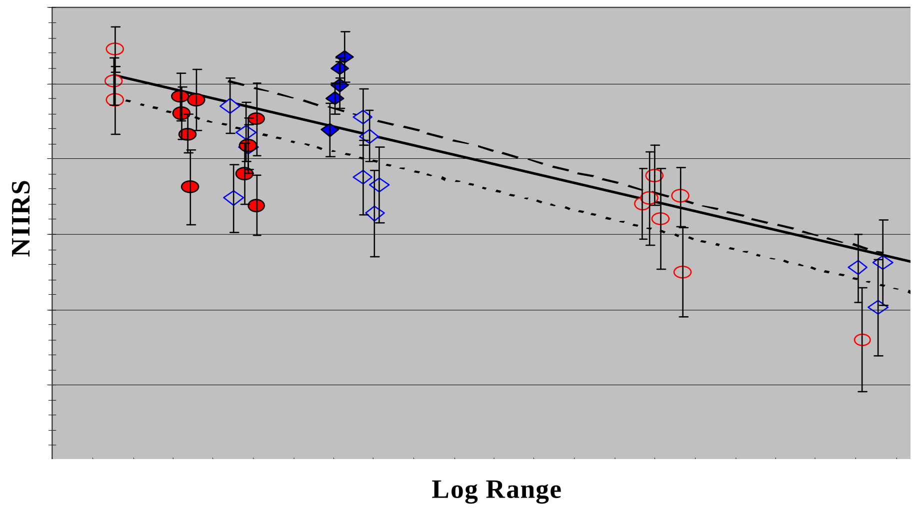

Figure 8 NIIRS Prediction Quality from the GIQE

For IR and EO sensors, generalizations may be made about image quality predictions. Thus, the GIQE NIIRS prediction for EO and IR sensors may be reduced to a function of geometry alone with reasonable accuracy. This is the basis for NIIRSE calculations made by the IFRS. Due to the sensitive nature of quantitative system performance data, only representative values are used in the IFRS. The general trend is shown in Figure 9 [11]. Less is known about image quality prediction for SAR. It is generally highly nonlinear with respect to geometry and many other variables. Much effort is currently being put into quantifying and validating SAR image quality predictions, but it is largely classified and will not be addressed here [21].

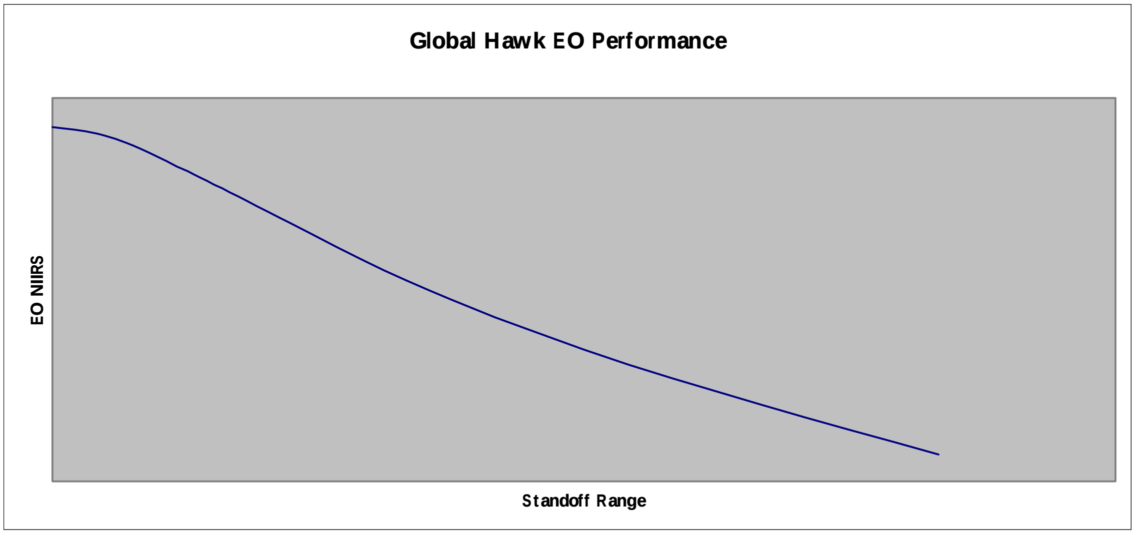

Figure 9 Qualitative NIIRS Degradation vs. Standoff Range: EO Sensor

3.4.2 Assumptions

It is a simple matter to transfer data such as route points, target locations, threat areas and types, wind direction, etc. between programs. Therefore, the main emphasis of the IFRS is to demonstrate the replanning function and not account for all details necessary for an operational system.

Global Hawk turns at cruise altitude are quite large in radius, normally at a standard radius of around 8 NM. Evasive maneuvering is obviously not possible, considering the thin atmosphere found at 60,000 ft. Even so, the relatively large map scale appropriate for the IFRS in these examples does not justify the computation of curved turns. Thus, turns are shown as vertex points in the IFRS as well as in the master mission plan.

The majority of Global Hawk missions are spent in 'cruise-climb' mode, defined as maintenance of an altitude range usually between 60,000 ft and 65,000 ft, depending on the mission. As fuel is burned, the air vehicle is allowed to creep upwards in altitude as it burns fuel and decreases in weight. The rate of climb during cruise-climb varies, but is generally less than 5 ft/min [20]. This further increases sensing range while putting more altitude between threats on the ground or from hostile aircraft, most of which service ceiling is well below 60,000 ft. Thus, altitude is assumed constant for the short segments of routes (around 30 minutes or less) being replanned in the IFRS.

Each of Global Hawk's sensors may operate in either spot or search mode. The IFRS allows for use of spot mode only to limit the scope of this research. Field of Regard (FOR) constraints (Figure 10/ Figure 11) limit the directions that the sensors may 'look' from the aircraft. For the EO and IR sensors, azimuth is constrained to +/- 15 degrees off each wingtip. The minimum time duration to collect images is also specified. This time value is specific to each sensor. HAE UAV CONOPS specifications were used for both imaging times and FOR values. Each point defining the current route is evaluated for eligible imaging locations. For imaging to be possible at any given route location, several criteria must be met; see Table 1.

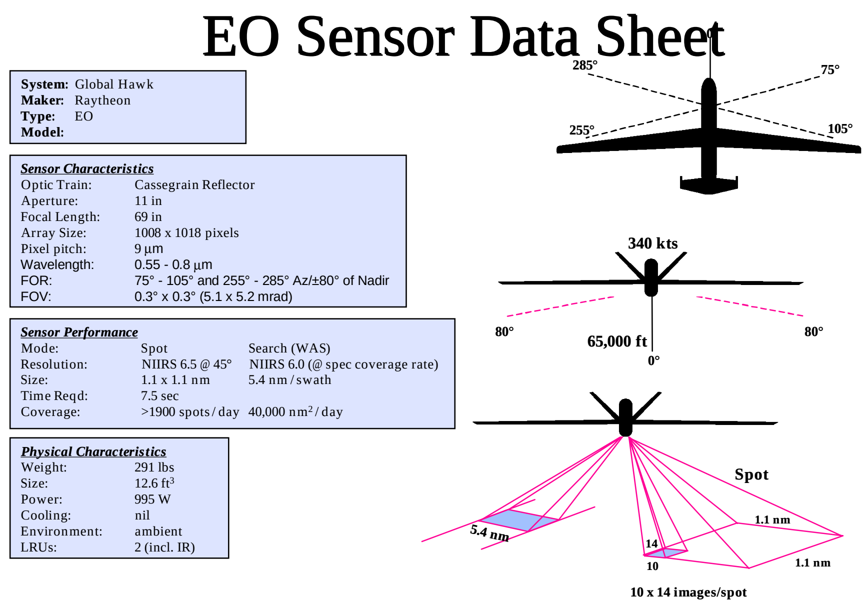

Figure 10 Global Hawk Electro-Optical Sensor Specifications [10]

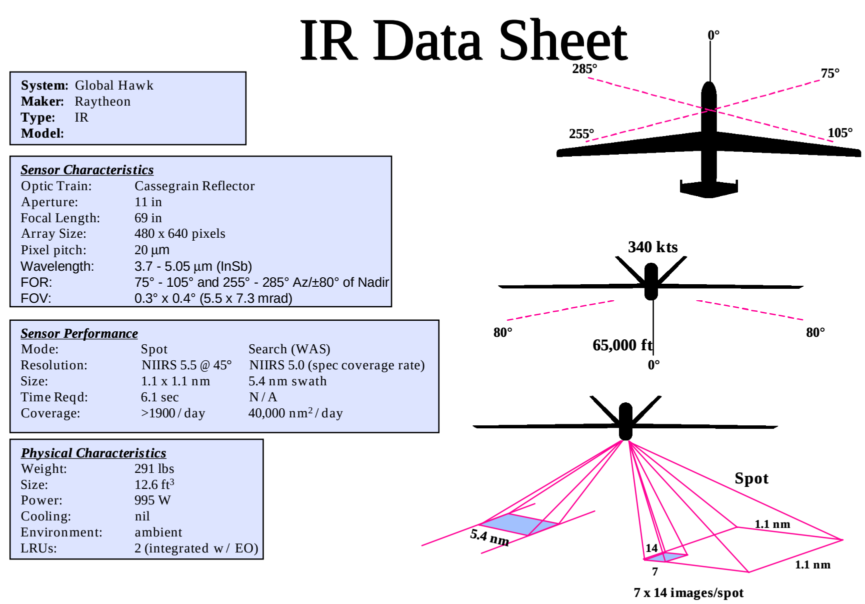

Figure 11 Global Hawk Infrared Sensor Specifications [10]

Table 1 Criteria Determining Valid Route Points for Imaging

• Aircraft not turning • Target within FOR limits • Sufficient time available within these constraints to complete image collection

3.4.2.1 Risk Definition

Sophisticated and highly classified threat models may be used in an operational environment to characterize the risk associated from particular threats, i.e. the SA-5 or SA-10 surface-to-air missiles, with high engagement envelopes relevant to Global Hawk missions. These models generate risk as a function of many variables, and are beyond the scope of this demonstration software. Risk from a single threat, $Risk_i$, is assumed to be a function of radial distance and time dwell within range of the hostile fire control radar according to the following equation, where *Cr = cost of risk factor* and *d = radial distance to threat:* $Risk_i = \int\left(Cr \cdot \frac{1}{d^2}\right)dt$ (unit risk) [24]. **Route Survivability** is a qualitative metric simulating a complex probability of survival calculation over the route segment in question, as in an operational system; there it would be calculated utilizing actual threat models combined with Monte Carlo runs. It is calculated here for demonstrational purposes only, with $Risk_I$ defined above and arbitrary *Weight:*

$$RouteSurvivability = \frac{1}{\sum Risk_i} \cdot Weight \quad .$$

3.4.3 Coordinate Systems

Three different coordinate systems are used in the algorithm. The WGS-84 Geodetic frame is used to interface with map coordinates. A local East-North-Up (ENU) frame is used for most position calculations. An earth-centered, earth-fixed (ECEF) rectangular Cartesian frame is used for calculating relative positions where additional precision is required, i.e. travel time calculations [14].

3.4.4 The Graphical User Interface

Figure 12 shows a route segment from the main OPUS mission displayed in the IFRS to start the iterative replanning process. This exocentric, or 'bird's eye' viewpoint is dominant among mission planning route displays where operators must frequently solve navigational problems and compare solutions with external sources. It is more efficient for the human operator if all information sources have a *consistent frame of reference*. For example, it's easier to compare map information between two earth-referenced, north-up displays than between one north-up map and one that rotates with an aircraft heading. Rapidly reorienting between frames of reference requires attention and can be difficult. If reference frames are consistent, mental transformation between frames is unnecessary in order to fuse the data into a meaningful picture [23]. This is critically important when rapid decisions must be made in stressful, in-flight environments where coordination occurs with ground forces, external mission planning elements, or other aircraft [18]. The user forms a new route by moving existing waypoints or adding additional ones using the **Adjust Route** and **Create Route** buttons described in section 3.4.

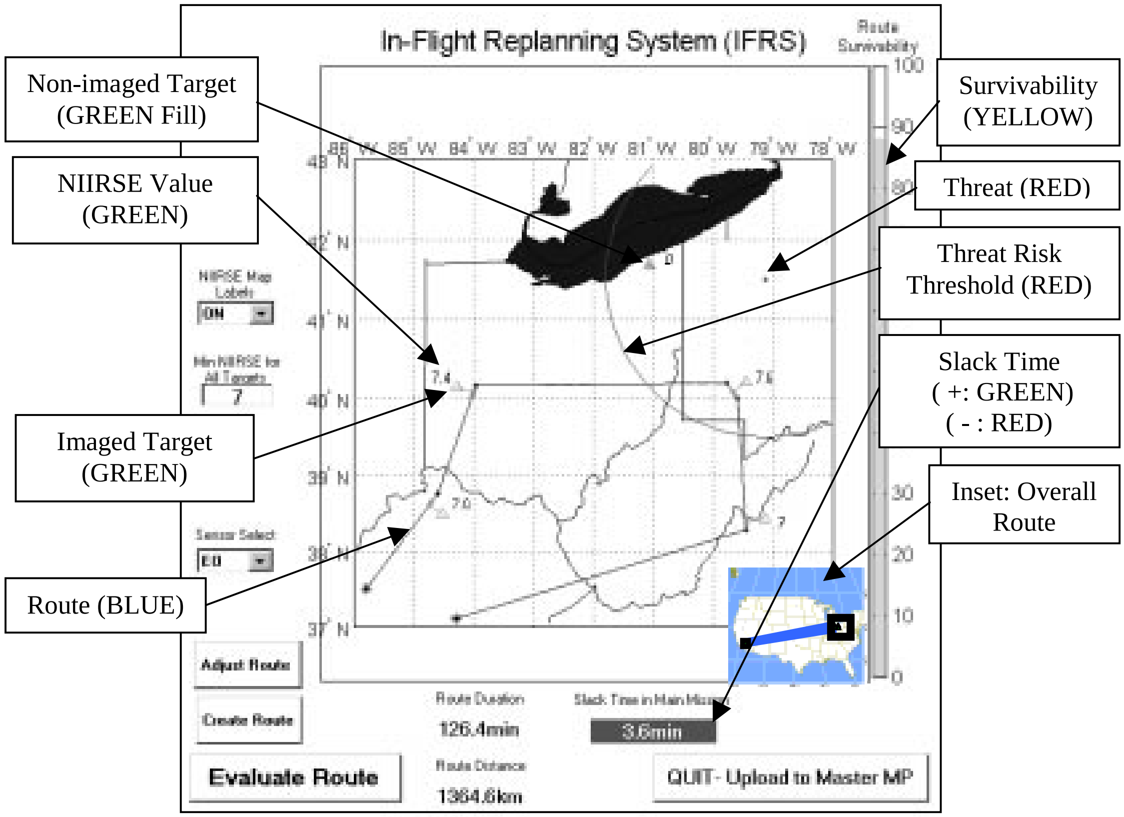

Figure 12 The IFRS GUI

A great deal of information is conveyed to the user about the route segment shown in Figure 12. To assist in data absorption efficiency, color is used extensively. Among visual display elements, color has been shown to hold short-term memory better than shapes or numbers. Memory is an important element in iterative revision of the route. Color processing is also fairly automatic and does not require much attention to recognize. It groups GUI attributes into larger categories more efficiently handled by short-term memory [23]. Green connotes 'good' and 'in progress' status. Targets are displayed as green triangle symbols, with newly added targets filled in with green to stand out from preexisting targets. Green is used to denote all information relating to targets, such as NIIRSE values and path regions when imaging may take place. Red dots depict threat locations; newly added threats are overlaid with a small red circle. Threats are encircled with a ring depicting a risk threshold (reduced probability of survival), based on threat models for that threat type. Red and yellow connote danger and caution, respectively. All threat information such as the threat risk threshold rings and threat locations is displayed in red, while the **Route Survivability** meter is marked in yellow. See Table 2 for a summary of color use in the IFRS GUI.

Table 2 GUI Attribute Color Groupings

GUI Attribute Set	Associated Color on GUI
Threat Information:	
• Threat location	RED
• Risk threshold	RED
• Route Survivability	YELLOW
Target Information:	
• Target location	GREEN
• Route portions where imaging is possible	GREEN
• Route portions where active imaging is taking place	GREEN
• NIIRSE values on map	GREEN
• ‘Min NIIRSE’ text box value	GREEN
Route Information:	
• Route path	BLUE
• Turn points	BLUE

Portions of the route passing by a target during which imaging could be taking place are marked green. A green radial line to the target marks the optimal image collection location, provided the minimum NIIRSE specification is met. A minimum NIIRSE specification results from the user entering a value in the **Min NIIRSE for All Targets** editable text field. This assists the user in establishing a path satisfying an image quality 'floor' during route adjustment. The NIIRSE value is displayed in green near each imaged target, with a zero value indicating image quality does not meet the minimum NIIRSE. All target NIIRSE labels may be removed from the display by switching off the **NIIRSE Map Labels** pull-down selector if the map becomes too cluttered. In this case, the NIIRSE value for a specific target will be displayed in a small pop-up window that appears when the mouse pointer pauses momentarily over a target. If targets must be viewed from a restricted angle (as in to survey the dry side of a levee), green arcs are drawn around them at the radius that would provide images satisfying the minimum NIIRSE specification. Duration and distance of the route segment being modified are displayed in blue, as well as the path itself. If positive slack time is available in the current route, the **Slack Time** value is highlighted in green. Likewise if a future TOT constraint will not be met with the current route (i.e. negative **Slack Time**), the value is highlighted in red and displayed as a negative value. Also shown in the IFRS GUI is an inset of the master mission plan. Once the segment being replanned with the IFRS is finished, the user presses the **QUIT-Upload to Master MP** button to terminate the IFRS application and insert the new segment back into the master mission plan. The remainder of the mission will then be reoptimized as time allows, using the full-featured master in-flight replanner.

3.5 IFRS Application

3.5.1 Basic Application: No Threat Considerations

Let’s revisit the Missouri flood area scenario to see how the IFRS would be employed. First of all, the mission would be completely laid out in advance of departure, just as missions are planned today. The mission executes normally right up to when the call comes in to the MCE for emergency assistance at the nearby town area. At this point, it becomes apparent to the mission commander that the mission plan needs to be altered to do some ad hoc surveillance in support of the local search crews. To start the process, the mission planning officer downloads the local region and the included mission plan segment to the IFRS (Figure 13). They will be modifying a 30-minute (depending on the size of the route affected) portion of the route beginning 5 minutes (determined by the time available before reaching the route segment being replanned) from Global Hawk's current position. The **Slack Time** field displays time remaining within the levee analysis optimal time window after sundown. Coordinates for the town's new image target (a filled-in green triangle) are automatically added to the mission plan database and displayed on the map with the original route through the area. The route also shows where images of preexisting targets (hollow green triangles) would be collected if the mission were left unchanged. The minimum image quality required for levee analysis is a NIIRS of 6.5, so the **Min NIIRSE for All Targets** editable text field has been set accordingly. Side-on views (as opposed to directly overhead) of the levee present the best geometry for analysis. This means the route should yield images greater than 6.5 NIIRSE but remain as far away from the target as possible for side viewing.

Since no threats are present for this mission, the **Route Survivability** meter is fixed at 100% to suggest it should be ignored.

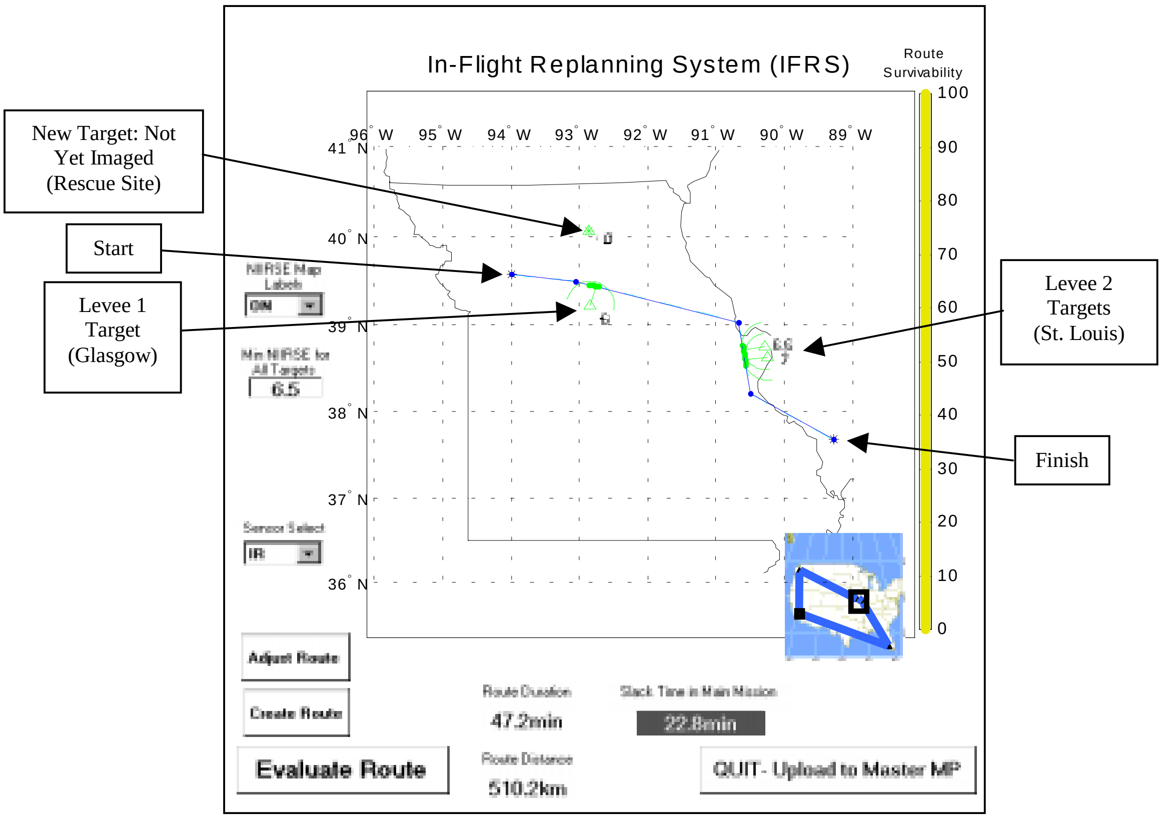

Figure 13 Example 1:
Mission Segment Downloaded to IFRS with New Target

Once the mission commander receives authorization to deviate from the original mission plan, the decision is made to reroute to the rescue area but also collect an image of the now distant, first levee (Figure 14). The **Min NIIRSE for All Targets** editable text field has been reset to 4 to help plan the route shown, yielding an interpretable but sub optimal 4.3 NIIRSE picture of the first levee.

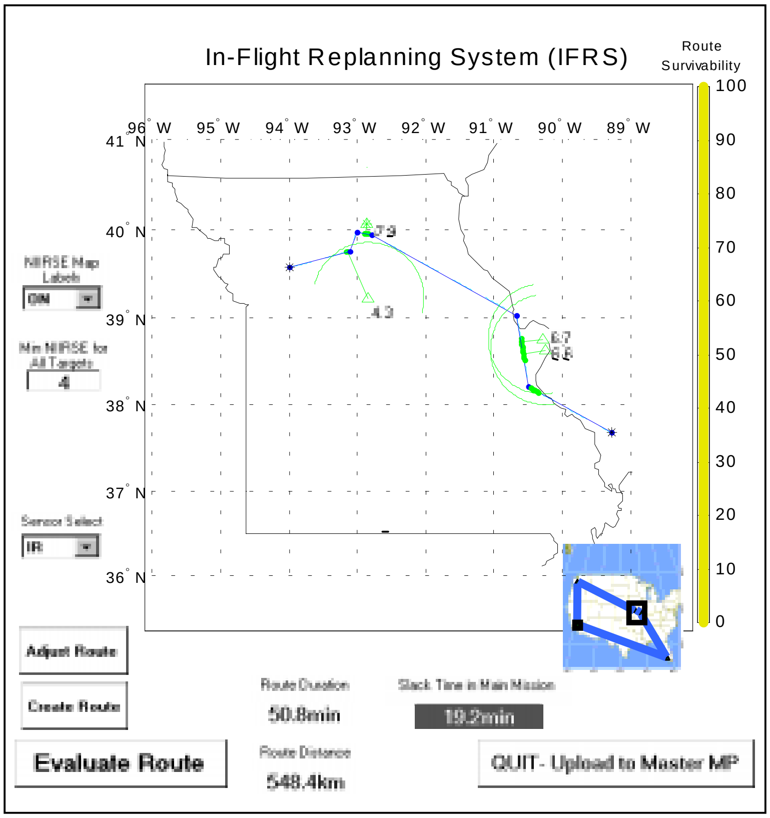

Figure 14 First Replan: Image the Town Rescue Area

It is clear from cursory interpretation of the levee scans, now imaged at too great a distance for accurate analysis, that the levees are in bad shape. The weak condition of the Glasgow levee, the unknown condition of the St. Louis levee, the 24 hours before another imaging opportunity, and the immediacy of the rescue operation support all must be weighed quickly. The decision is made to break off from the rescue operation, leaving them with at least a single survey pass of their search area. A new route is defined to the first levee for better pictures, continuing on to the second levee area in St. Louis with only 7.2 minutes of **Slack Time** within the optimum levee survey time window (Figure 15).

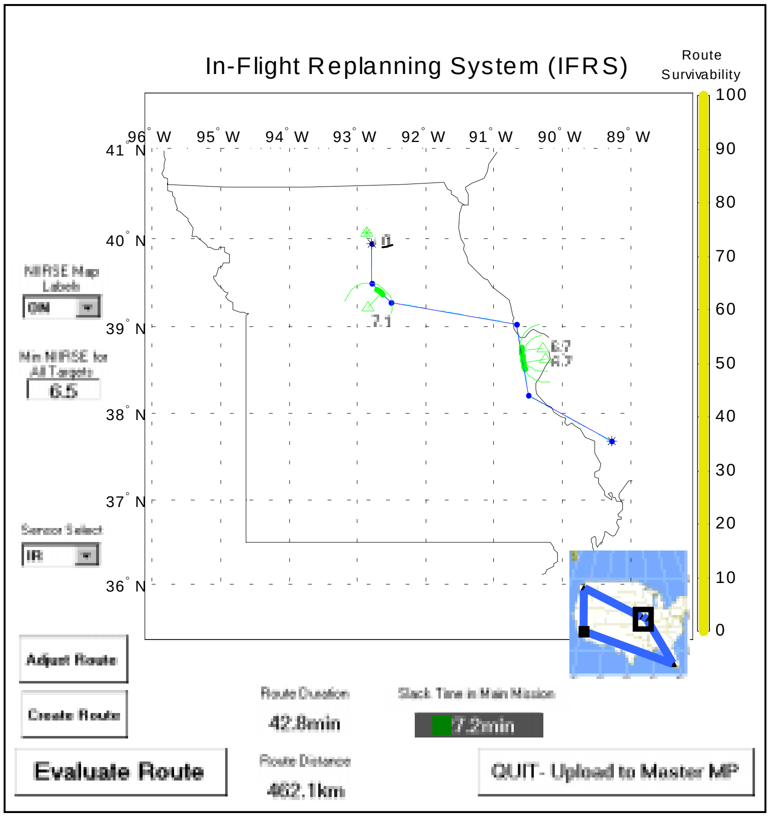

Figure 15 Second Replan: Re-image First Levee at Optimum Distance

3.5.2 Full IFRS Capability Application: Scenarios With Threats

To demonstrate the capability of the IFRS in a threat environment, let's revisit the first example of section 3.2.2. It specifies the addition of a priority pop-up target to the immediate area [16]. The target is a parking lot however, and only a count and general classification of the vehicles is required. It’s determined that an EO NIIRS of 5 or greater will fulfill the requirement. The 4 preexisting targets in the area are antiaircraft artillery (AAA) batteries suspected to be of the newest type that have been lethal to Army Apaches operating in the area. NIIRS ratings of 7 or greater are deemed necessary for type classification of the AAA guns. While AAA is not a threat to the high flying Global

Hawk, a SA-10 surface-to-air missile (SAM) emplacement to the northeast protects the area. An added consideration for the mission commander is a TOT (Time on Target) constraint: the planned upcoming surveillance of a road intersection where a covert terrorist meeting is supposed to take place at a given time later in the mission. This constraint leaves little slack time to lengthen the route.

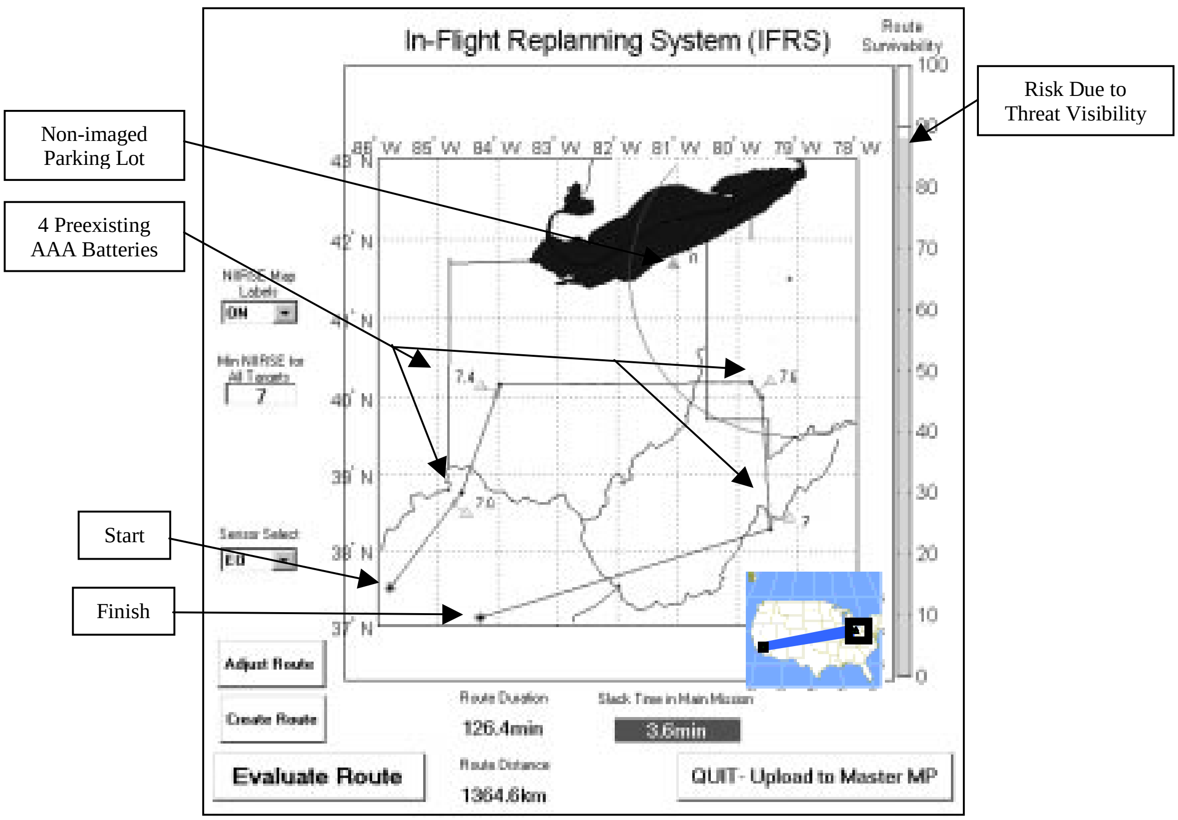

Figure 16 Example 2:
Mission Segment Downloaded to IFRS with New Target

The challenge is to quickly reroute for the new scenes, collecting images meeting the required minimum NIIRS ratings and avoiding the SAM threat as much as possible, while weighing the covert meeting TOT. Figure 16 depicts the initial IFRS screen with new target before rerouting.

The mission commander may decide the terrorist meeting is somewhat lower priority and hedges a bet that it will take place towards the end of the time window. All targets are imaged with greater than the minimum NIIRS requirements at a cost of missing the first 15 minutes of the TOT time window and increased threat risk. Figure 17 shows the resulting replan solution. It is then uploaded back to the master mission plan, which reoptimizes the remaining mission route.

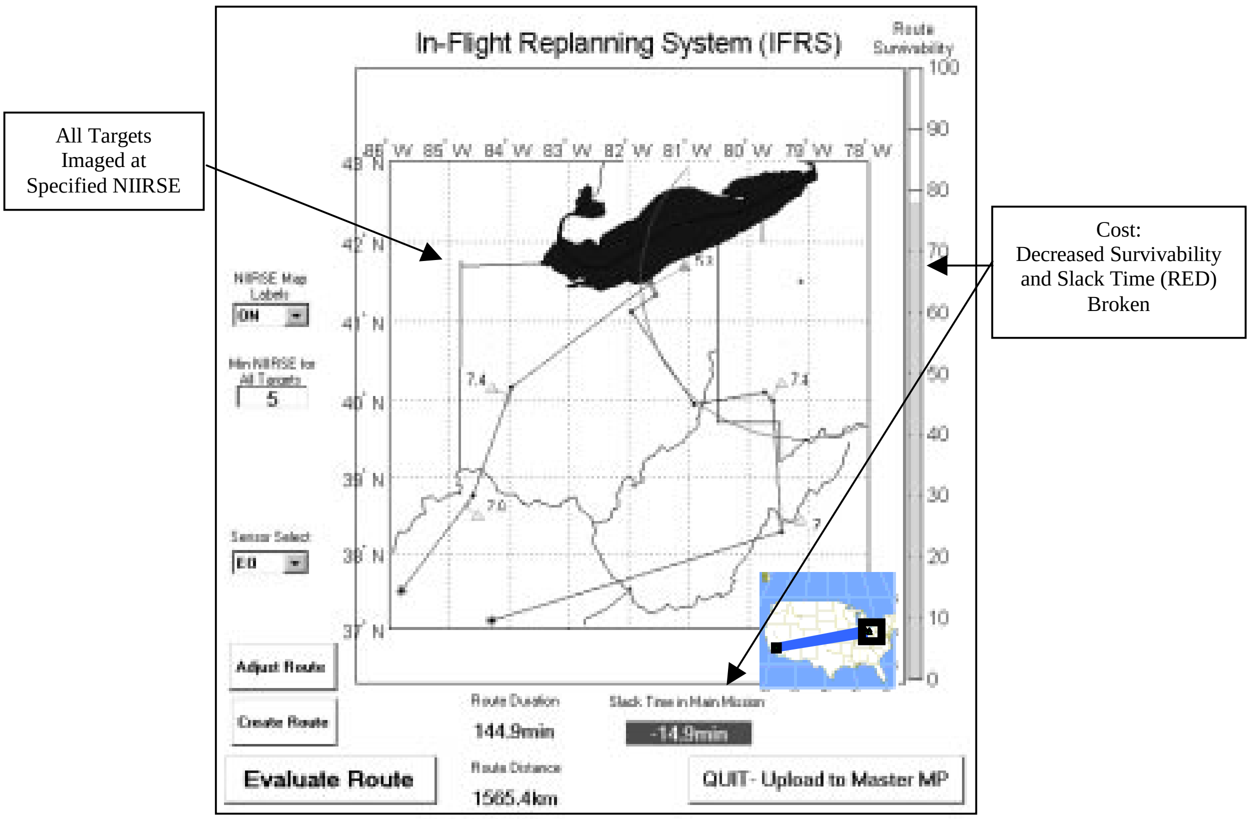

Figure 17 Replan Option 1: Break TOT Constraint

Another option for the mission commander is to accept reduced image quality for the 2 east most targets and make the collection anyway. Figure 18 shows this option.

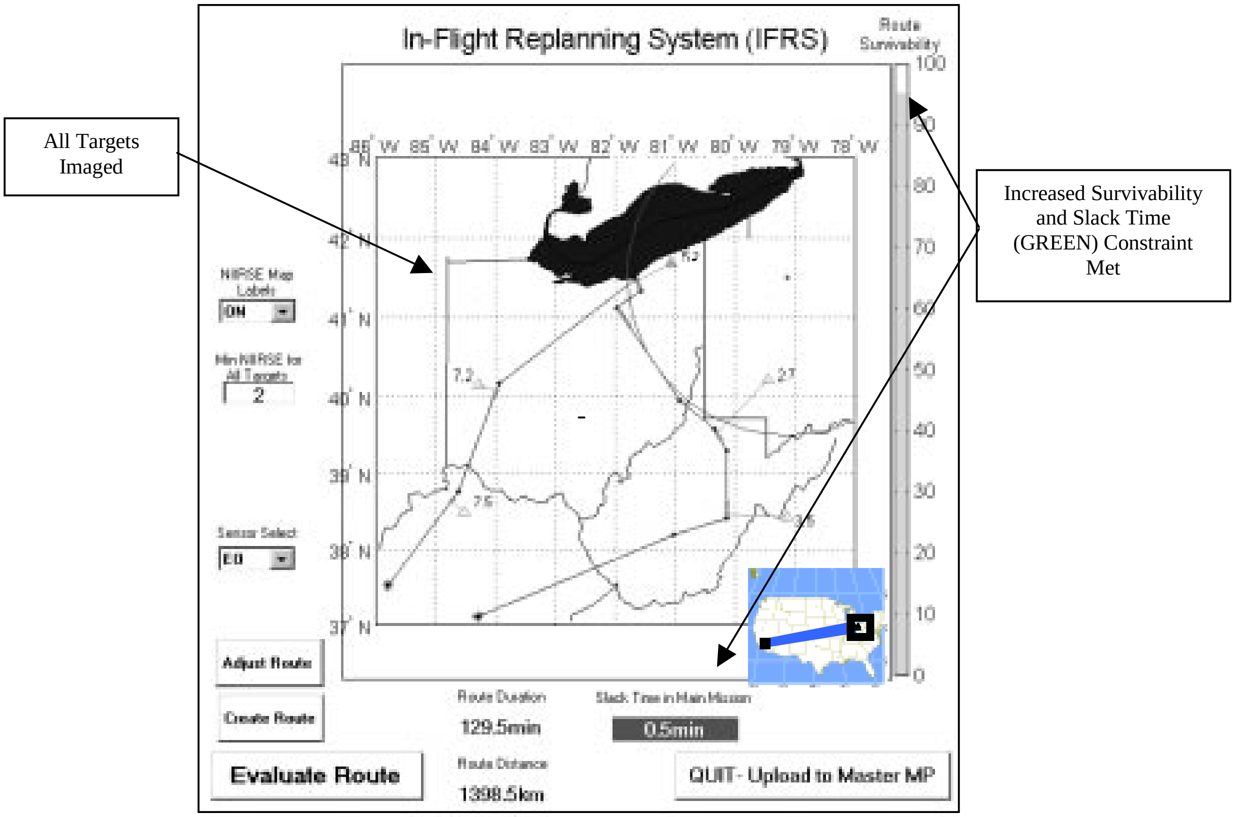

Figure 18 Replan Option 2: Break NIIRSE Specification for 2 Targets

3.6 The Future for In-Flight Mission Planning

The state of the art for in-flight mission replanning is rapidly advancing. The advent of broadband communications and ultra-high speed data rates has made supervisory-controlled UAVs like Global Hawk a reality, as well as the near-real-time transmission of their images. Much more is on the horizon however, like ‘smart’ replanners utilizing fuzzy logic or neural networks capable of analyzing contingencies and developing solutions on a much deeper level than is now possible. Other in-flight technologies, like the Rotorcraft Pilot’s Associate currently being developed for the Army by Boeing Mesa, demonstrate in the AH-64D Apache Longbow the next generation of mission planning capability. “The system is smarter than many pilots and faster than the best,” said CW4 John E. Vandenberg, Army RPA chief test pilot. The system handles sensor data from on-board radar, off-board sources via the Improved Data Modem (IDM), Joint Tactical Information Distribution System (JTIDS), radio frequency interferometery (RFI) from battlefield threat emitters, infrared targeting and acquisition system (TAS), and on and on. Huge quantities of battlefield data are fused for predicting target motion, providing fire control for other weapons platforms, maneuvering evasively around pop-up threats, etc. The human interface is highly advanced with multiple-display visual and 3-D auditory cues. Eventually, these advanced capabilities will be assimilated into the UAV arena, further emphasizing the hands-on role of the operator [5].

4 Conclusions and Recommendations

4.1 Conclusions

The IFRS allows users to take into account many properties of mission planning that are difficult to quantify. As such, human operators are used in their advantageous capacity to weigh choices and determine an acceptable result. This is in contrast to the conventional optimization process that seeks to find a 'best' solution; often we seek a solution that will simply **work**, given frequently time-limited and dynamic operational environments.

4.2 Recommendations and Further Research Opportunities

The first step toward implementation of IFRS-like in-flight replanning tools is to continue refining the HAE UAV conventional mission planning process. Mission planning duration must be shortened and the process simplified; the bimonthly Global Hawk Mission Planning Working Group (MPWG) meetings are one example of this monumental effort well underway. No doubt the mission planning streamlining effort will continue for some time.

Next, this advanced mission planning capability must be extended to real time, on-the-fly control of missions during execution. Eventually, the full capability of preflight mission planning will be available to the MCE in flight. This capability will then need to be further streamlined into a simplified in-flight mission replanner like the IFRS for use in time-critical replanning scenarios. Of course, the application of a full-featured preflight mission planning system to in-flight mission replanning is not a one step procedure. It will evolve over time, gradually building capability. In the real world of tight budgets, aggressive schedules, milestone counting, and report/paperwork generation, development of the conventional and in-flight mission planning processes outlined in this thesis must occur simultaneously. Thus, we recognize that the real world evolution of in-flight mission replanning is more complicated and convoluted than an ideal on paper.

Opportunities for further research are present for investigating the impact of variable degrees of automation on various replanning tasks. An IFRS-like tool containing some level of path optimization to assist the operator during path replanning could be developed. Human subject trials could be conducted to further understanding of which combinations of tasks, time limits, and automation are most effective.

4.3 Final Remarks

In 1996, the Air Force Chief of Staff directed the Air Force Scientific Advisory Board (SAB) to conduct the study, “UAV Technologies and Combat Operations”. Among their findings were:

- UAVs have significant potential to enhance the ability of the Air Force to project combat power in the air war.
- UAVs have the ability (range, persistence, survivability, and altitude) to provide significant surveillance and observation data economically, compared with current manned aircraft approaches.

- UAVs have the potential to accomplish tasks that are now, for either survivability or other reasons, difficult for manned aircraft including counterair (cratering runways and attacking aircraft shelters), destroying or functionally killing chemical warfare/biological warfare (CW/BW) manufacturing and storage facilities, and suppression of enemy air defenses (SEAD).

- Insufficient emphasis has been placed on human systems issues. Particularly deficient are applications of systematic approaches to allocating functions between humans and automation, and the application of human factors principles in system design. [1]

UAVs are proving their worth with positive operational experiences, such as the previewing of CAP areas and targets for F-16 pilots in Bosnia [18]. Each success story gains a few more supporters, especially when a significant operational impact is made. Despite the growing pains associated with a relatively new, fast moving technology, UAVs and the mission planning systems that control them are gaining a foothold in the operational world; a stepping-off point into the battlefield of the 21st century.

Appendix A: IFRS Matlab Code

```
% Thesis Main Code, Capt Dave Pritchard, AFIT GAE-00M-10
% Version 3.33, Build 3
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This script requires the following .m files, which
%   are executed as subfunctions:
%%  setup333.m
%%  ring225.m
%%  arc225.m
%%  rad2meter.m
%%  lla2ecef.m
%%  crad2heading.m
%%  checkpath.m
%%  create.m
%%  adjust.m
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Coordinate system denotation conventions:
%%  xx_g = Geodetic frame (WGS-84 unless o/w stated)
%%       = [latitude longitude altitude]
%
%%  xx_gd = [deg deg m]
%%  xx_gr = [rad rad m]
%%  xx_gm = [m   m   m]  (not true geodetic but local ENU: named for
convenience)
%%%%%%
%%  xx_e = Earth Centered, Earth Fixed frame
%%       = [m  m  m]
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Task Switches: switch on or off associated features
%  X_sw =1 for 'on'
%  X_sw = 0 for 'off'

TargVisArcs_sw = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% INITIALIZATIONS:

% sensortype:(1=EO 2=IR)
%sensortype = 1;
% Convert min NIIRSE Specification to min Ground Range = [km]
if MinNiirse == 0,
  grspec = inf;
else,
  if sensortype == 1,
    grspec = interp1(EOdata(:,2),EOdata(:,1),MinNiirse);
  elseif sensortype ==2,
```

```
    grspec = interp1(IRdata(:,2),IRdata(:,1),MinNiirse);
  else,
    disp('Error: Unexpected sensortype'),
  end,
end,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% x,y = uninterpolated values = xuser, yuser = [deg]
% fill in between selected route points for increased resolution
maxx = max(xuser);
minx = min(xuser);
maxy = max(yuser);
miny = min(yuser);
% 'maxdiff' = [deg] sets the largest acceptable lat or lon increment
%   filling-in path with linear interpolation:
% divides smallest change in lat or lon (endpt-startpt) by resolution
maxdiff = min([(maxx - minx)/resolution (maxy - miny)/resolution]);
maxdiff_deg = maxdiff;
maxdiff_deg,
segment =[]; xfit =[]; yfit =[];
for p = 1:(length(xuser)-1), % pre-interpolated # of path pts.
  xtemp = []; ytemp = [];
  [ytemp, xtemp] = interpm(yuser(p:p+1),xuser(p:p+1),maxdiff);
  % build x,y columns of path1
  xfit = [xfit ; xtemp];
  yfit = [yfit ; ytemp];
  % build 5th column of path1 = [path segment number]
  segment = [segment(:) ; p*ones(length(xtemp),1)];
end
% update x,y = [deg] to interpolated values
x = xfit;
y = yfit;
% update npath1 to # pts post interpolation
npath1 = length(x);
% fill in z1 (altitude), v1 (flt. velocity)
% z=const=20km, npath1 pts
alt = 20000; %[m]
vel = 180;   %[m/s]  180 m/s = 350 kts
z1 = ones(npath1,1)*alt;
% velocity at each n pts
v1 = ones(npath1,1)*vel;
% clear axes & plot previous path iteration
axes(mapH);
cla,
axesm('mapprojection','mercator',...
  'maplatlimit',maplatlimit,'maplonlimit',maplonlimit),
patchm(uslat,uslon,mapbackgrndclr),
patchm(gtlakelat,gtlakelon,[0 0 .75]),
plotm(statelat,statelon,'k'),
gridm('mlinelocation',1,'plinelocation',1),
mlabel('mlabellocation',1);
plabel('plabellocation',1);
hold on,
% 6th column for path1 = noturn(y/n) = [binary]
sega =[]; segb =[]; yesturn =[]; noturn =[];
noturn = ones(npath1,1);
```

```
segb = segment(2:npath1,1);
sega = segment(1:npath1-1,1);
yesturn = segb - sega;
% add back initial point and assume it's in a turn
yesturn = [1; yesturn]; % now length = npath1 again
% make noturn = 0 for vertex points
noturn = noturn - yesturn;
% make last point a turn point
noturn(npath1,1) = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% create new path1 (length(x)) for evaluation
% initialize path1 matricies
% note: path1_gx and path1_e all contain velocity and seg# columns
path1_gd =[]; path1_gr =[]; path1_e =[];
%%%%%%%%
path1_gd = [y,x,z1,v1,segment,noturn];  % geodetic frame [lat lon alt]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% compute course heading
% note: length(heading1) = (npath1 - 1)
pathb_g = path1_gd(2:npath1,1:2);
patha_g = path1_gd(1:(npath1-1),1:2);
dheading = pathb_g - patha_g;
% 'heading1_r' = [radians] on interval (-pi,pi) w/ x-axis reference
heading1_r = atan2(dheading(:,1),dheading(:,2));
% 'heading1' = [deg] on interval (0,360) w/ true-North reference
heading1 = crad2heading(heading1_r);
% convert deg to rads for lla2ecef.m
path1_gr = [path1_gd(:,1:2).*pi/180 path1_gd(:,3:5)];
% convert to ECEF coords for analysis
path1_e = lla2ecef(path1_gr(:,2),path1_gr(:,1),path1_gr(:,3)); % ECEF
frame [m]
path1_e(:,4:5) = path1_gr(:,4:5);
if initial_run == 0,
  % plot previous path for comparison
  path0_gd = path1_gd;
  plotm(path0_gd(:,1),path0_gd(:,2),'c--'),
end,
%%%%%%%%%% uncomment below to see individual interpolated path pts.
%plotm(path1_gd(:,1),path1_gd(:,2),'b.'),
% plot new route
plotm(yuser,xuser,'b'),
plotm(yuser,xuser,'b.'),
% plot first and last pts black
plotm(path1_gd(1,1:2),'k*'),
plotm(path1_gd(npath1,1:2),'k*'),
% compute travel time and distance
dist1 =0; dist_i =0; dpath =[]; time1 =0; path2 =[]; path1 =[];
% subtract (i+1) shifted path w/ (i) path
% length of each is (npath1-1)
pathb_e = path1_e(2:npath1,1:3);
patha_e = path1_e(1:(npath1-1),1:3);
dpath_vec = pathb_e - patha_e;
% norm across rows
dpath = sqrt(dpath_vec(:,1).^2 + dpath_vec(:,2).^2 +
dpath_vec(:,3).^2);
```

```
dist1 = sum(dpath);
time1_vec = dpath./path1_e(2:npath1,4);
time1 = sum(time1_vec);
format bank,
travel_distance_km = dist1/1000,
travel_distance_nm = dist1/1852,
travel_time_minutes = time1/60,
travel_time_hours = time1/3600,
format short,
% plot threat points
plotm(tht_gd(:,1),tht_gd(:,2),'r.'),
% plot threat rings at rmarr radius w/'ring.m'
% assumes ring radii are precalculated for current flight altitude
(20km)
% actually, project a/c path to ground: same effect; simpler!
for i = 1:length(tht_gd(:,1)),
  ring225(tht_gd(i,1:2),rmarr(i),'r')
end,
% compute threat metric: 'risk' ~ 1/r^2
% assumes ring radii are precalculated for current flight altitude
(20km)
% metric based on 2-D map projection of flight path to grnd level
dtht =[]; rad2tht =[]; risk = [];
path1_gnd_gr =[]; path1_gnd_e =[]; dtht =[];
ththits = ones(length(tht_gr(:,1)),1);
risk    =zeros(length(tht_gr(:,1)),1);
% 't' steps through each threat (each row of 'tht', threat 't')
for t = 1:length(tht_gr(:,1)),
  % project path1 down to 0km (ground level)
  path1_gnd_gr = [path1_gr(:,1:2) thtalt*ones(npath1,1)];
  path1_gnd_e =
lla2ecef(path1_gnd_gr(:,2),path1_gnd_gr(:,1),path1_gnd_gr(:,3));
  % vector distance to threat for each path point
  dtht = path1_gnd_e(:,1:3) - ones(npath1,1)*tht_e(t,1:3);
  % radius to threat for each path point
  rad2tht = sqrt(dtht(:,1).^2 + dtht(:,2).^2 + dtht(:,3).^2); % =[m]
  % 'p' sums all radii to threat < rmarr
    % initialize counters
  ththitsseg = 0;
  riskseg = 0;
  oldseg = path1_gd(1,5);
  for p = 1:npath1,
   % test if inside risk area
   if rad2tht(p) <= rmarr(t) %[m]
     currentseg = path1_gd(p,5);
     % test if route segment has changed
     if currentseg == oldseg,
       % count threat hits inside rmarr while route segment unchanged
       ththitsseg = ththitsseg + 1;
       riskseg = riskseg + (1/(rad2tht(p)/rmarr(t)))^2;
       if p == npath1,
         % normalize riskseg by # of ththits counted
         % this accounts for varying interpolation density btw.
segments
         % trap div by zero
```

```
          riskseg = riskseg/ththitsseg;
          % add to risk running tally
          risk(t) = risk(t) + riskseg;
          % reset counters
          ththitsseg = 0;
          riskseg = 0;
        end, %if p == npath1
      else, %change seg # inside a tht region
        % normalize riskseg by # of ththits counted
        % this accounts for varying interpolation density btw. segments
        riskseg = riskseg/ththitsseg;
        % add to risk running tally
        risk(t) = risk(t) + riskseg;
        %reset counters
        ththitsseg = 0;
        riskseg = 0;
      end, %if currentseg
           % test if left tht region w/o tallying risk up
    elseif ththitsseg,
      % normalize riskseg by # of ththits counted
      % this accounts for varying interpolation density btw. segments
        %if ththitsseg,
        % trap div by zero
        riskseg = riskseg/ththitsseg;
        % add to risk running tally
        risk(t) = risk(t) + riskseg;
        %reset counters
        ththitsseg = 0;
        riskseg = 0;
      end, %if rad2tht
      % update oldseg
      oldseg = path1_gd(p,5);
    end, %for p
end, %for t
if sum(risk),
  survivability = 1/(sum(risk))*survive_factor;
else,
  survivability = 100;
end,
% plot target points
plotm(targ_gd(1,1),  targ_gd(1,2),  'g^'),
if length(targ_gd(:,1)) >= 2,
plotm(targ_gd(2:length(targ_gd(:,1)),1),targ_gd(2:length(targ_gd(:,1)),
2),'g^'),
end,
% plot target visibility arcs
if TargVisArcs_sw,
  % assign vars for arc plotting function
  % center = [lat lon] = [deg deg]
  center = [targ_gd(:,1) targ_gd(:,2)];
  minlook = targ_gd(:,4);
  maxlook = targ_gd(:,5);
  arcrad = grspec*1000; % [m]
  % plot arcs
  arccolor = 'g';
```

```
  for i = 1:length(targ_gd(:,1));
    if 0,
      % min range arc
      arc225(center(i,:),minlook(i),maxlook(i),arcrad,arccolor)
    end,
    % max range arc
    arc225(center(i,:),minlook(i),maxlook(i),arcrad,arccolor)
  end, % target vis arcs
end,% TargVisArcs_sw
% compute EO/IR NIIRSE
% use lats, lons converted to [m] as local ENU cartesian frame (NIIRSE
DOP > this error)
%   this simplifies dist calcs, err < 350m all axes (see rad2kmtest.m)
% calculate local lon2m, lat2m: based on path1 start pt.
[lon2m, lat2m] = rad2meter(path1_gr(1,1),path1_gr(1,3));
% initial calcs for housekeeping
% convert path1 to local ENU frame (see above)
% path1_gm not really geodetic, but use nomenclature for consistency
% below is an example of improper use of rad2meter.m:
%   conversion is good only for relative distances, not absolute dists
%   path1_gm = [lat2m*path1_gr(:,1) lon2m*path1_gr(:,2)
path1_gr(:,3:4)];
Slant_Range2targ_min_nm = []; Ground_Range2targ_min_nm =[];
Slant_Range2targ_min_km = []; Ground_Range2targ_min_km =[];
el_min_deg =[]; az_min =[];
az2targ_check_deg_out = [];
% step through each target
for t = 1:length(targ_gr(:,1)),
  dtarg_gm_vec = []; dtarg_gm = [];
  gr2targ =[]; dh2targ =[]; forsr2targ =0; forgr2targ =0;
  az2targ =[]; el2targ =[];
  % 1st, calc relative position vector to targ from path1 [rad rad m
seg#]
  dtarg_gm_vec = [-path1_gr(:,1:2)+ones(npath1,1)*targ_gr(t,1:2) ...
      -path1_gr(:,3)+ones(npath1,1)*targ_gr(t,3) ...
      path1_gr(:,5)];
  % now convert relative position vector to [m m m seg#]
  dtarg_gm_vec(:,1:2) = [lat2m*dtarg_gm_vec(:,1)
lon2m*dtarg_gm_vec(:,2)];
  % slant range to current target
  sr2targ = sqrt(dtarg_gm_vec(:,1).^2 + dtarg_gm_vec(:,2).^2 +
dtarg_gm_vec(:,3).^2);
  % ground range to current target (w/ flat earth assumption)
  gr2targ = sqrt(dtarg_gm_vec(:,1).^2 + dtarg_gm_vec(:,2).^2);
  % take abs to remove neg sign depicting downward direction
  dh2targ = abs(dtarg_gm_vec(:,3));
  el2targ = acos(dh2targ./sr2targ);
  % get direction to targ, (excluding initial point to match heading1
dimension)
  %  heading2targ is heading TO targ(t) FROM path1(p,:) point
  % exclude initial point: heading1 is length = npath1-1
  heading2targ_r = atan2(dtarg_gm_vec(2:npath1,1),
dtarg_gm_vec(2:npath1,2));
  % get angle btw heading and targ az
  az2targ = abs(heading1_r - heading2targ_r); % vector, cart. ref [rad]
```

```
  % add back initial point making az2targ length npath1 (matches path1
again for calcs)
  az2targ = [0;az2targ];
  pathdata = [];
  pathdata = checkpath(path1_gd,npath1,az2targ,el2targ,sensortype);
  % store pathdatas from each targ in cell array
  pathdata_cell{1,t} = pathdata;
  step = 1;
  forcount = 1;
  validseg = 0;
  forcount_segvec =[];
  targimagepts_GR =[]; targimagepts_maxGR  =[]; imaging_ps =[];
  for p = 1:step:(npath1),
    % plot current path point (debug aid)
    %plotm(path1_gd(p,1),path1_gd(p,2),'b.')
    % while seg# unchanged, continue; o/w reset %oldseg =
path1_gr(p,5);
    % check for acceptable a/c FOR and get min dist to targ
    if path1_gd(p,6), % check if noturn = 1
      % check if valid image pt and above min NIIRSE specification
      if (sum(pathdata(p,:)) == length(pathdata(p,:)) &
(gr2targ(p,1)/1000 <= grspec)),
        % Calcs done in here denote FOR ok pts
        validseg = 1; % binary (1/0)
        forsr2targ(forcount,1) = sr2targ(p);
        forgr2targ(forcount,1) = gr2targ(p);
        % store record of valid p's w/ seg #
        recordp(forcount,:) = [p path1_gd(p,5)];
        % count # valid pts in current segment
        forcount = forcount + 1;
        % plot colored dots after actual imaging segment determined
        plotm(path1_gd(p,1),path1_gd(p,2),'g.'),
      end,
      % Below section for EO & IR: Do SAR separately
      % Must only allow single pt turns, o/w below fails (ie if
'turnplot.m' is implemented)
    else % If encounter a turn pt,
      if (validseg & p ~= 1),% If seg had valid image pts & skip
initial point,
        % plot turn point colored
        %plotm(path1_gd(p,1),path1_gd(p,2),'k*'),
        % now have a list of ground ranges
        % sort valid ground ranges
        % GRvalids has 2nd & 3rd columns = orig p-index value from
path1 and seg #
        GRvalids = sortrows([forgr2targ recordp]);
        % compute # pts req'd for image collection w/sensortype & round
up
        %   use time1 at 1st pt in group of valid imaging pts to
approximate time1 over whole range
        % trap div by zero in time1_vec if at a turn pt
        if time1_vec(GRvalids(1,2)),
          nptsreqd =
ceil(imagetime(sensortype)/time1_vec(GRvalids(1,2)));
        else,
```

```
          nptsreqd =
ceil(imagetime(sensortype)/time1_vec(GRvalids(1,2)+1));
        end, %if time1_vec
        % store range of req'd pts and associated p #'s
        % targimagepts_GR = [possible imaging ground ranges for this
segment]
        targimagepts_GR = GRvalids(1:nptsreqd,1);
        % segimaging_ps = [possible imaging p#'s for this segment]
        segimaging_ps = GRvalids(1:nptsreqd,2);
        % add seg#: segimaging_ps = [p seg#]
        % take seg# at point (p-1) to avoid taking updated seg# @ turn
pt
        segimaging_ps = [segimaging_ps ones(nptsreqd,1)*path1_gd(p-
1,5)];
        % add to running tally (entire path)
        % imaging_ps = [p seg#]
        imaging_ps = [imaging_ps ; segimaging_ps];
        % If npts > length of GRvalids, display error or move on
        % build vec of max valid image grnd rng from ea seg and
associated seg #
        % include seg # for plotting colored pts later
        targimagepts_maxGR = [targimagepts_maxGR; [max(targimagepts_GR)
path1_gd(p-1,5)]];
      end, %if validseg,
      % do below if at a turn pt, regardless of validseg
      % forcount_segvec: vector of # of valid image points per segment
      forcount_segvec = [forcount_segvec; forcount];
      % reset for next segment,
      forcount = 1;
      validseg = 0;
      segimaging_ps =[]; nptsreqd =[];
      recordp =[]; GRvalids =[];
      forsr2targ =[]; forgr2targ =[];
    end, %if noturn
  end, %for p
  % route analysis is now complete
  % before cycle to next targ, store values for current targ in cell
arrays:
  % targimagepts_maxGR = [(maxGR value) (associated seg#)]
  % sort to identify shortest GR image segment
  targimagepts_maxGR = sortrows(targimagepts_maxGR);
  % trap empty sets
  if targimagepts_maxGR,
    % store lowest value of max grnd range from each image segment
    Ground_Range2targ_min_km{t,1} = targimagepts_maxGR(1,1)/1000;
    % store seg# where imaging actually takes place
    imageseg = targimagepts_maxGR(1,2);
    % imaging_ps = [p seg#]
    % plot imaging_ps's only of segment 'imageseg'
    for i = 1:length(imaging_ps(:,1)),
      if imaging_ps(i,2) == imageseg,
        % plot line to actual image collection pts in color
        plotm([[path1_gd(imaging_ps(i,1),1:2)];[targ_gd(t,1:2)]],'g'),
      end, %if
    end, %for i
```

```
  end, %if (null trap)
  forcount_out{t,1} = forcount_segvec;
  %Slant_Range2targ_min_nm(t,1) = min(forsr2targ(:,1))/1852;
  %Slant_Range2targ_min_km(t,1) = (min(forsr2targ(:,1))./1000);
  %Ground_Range2targ_min_nm(t,1) = (min(forgr2targ(:,1))./1852);
  %Ground_Range2targ_min_km(t,1) = (min(forgr2targ(:,1))./1000);
  %az2targ_check_deg_out{t,1} = az2targ_check_deg;
  %el2targ_min_deg(t,1) = min(el2targ)*180/pi;
end, %for t
% now have cell ary of minGR2targ for ea targ
% interp for niirse using sensortype
% outputs to cmd line section:
% keeps track of # of valid FOR pts for ea. targ.
forcount_out,
Ground_Range2targ_min_km,
%Slant_Range2targ_min_nm,
%Ground_Range2targ_min_nm,
%Slant_Range2targ_min_km,
%Ground_Range2targ_min_km,
%el2targ_min_deg,
%az2targ_check_deg_out,
% Compute NIIRSEs
GR_niirse = Ground_Range2targ_min_km;
% throw out zero ground range values
niirse = zeros(length(targ_gd(:,1)),1);
% compute niirse for selected sensor and trap nulls
for t = 1:length(Ground_Range2targ_min_km),
  if (~ isempty(Ground_Range2targ_min_km{t})) & sensortype == 1,
    niirse(t) = interp1(EOdata(:,1),EOdata(:,2),GR_niirse{t});
  elseif (~ isempty(Ground_Range2targ_min_km{t})) & sensortype == 2,
    niirse(t) = interp1(IRdata(:,1),IRdata(:,2),GR_niirse{t});
  end, %if sensortype
end, %for t
% truncate niirse after one decimal place
niirse = round(10.*niirse)./10;
niirse,
initial_run = 0;
% Update target tooltip NIIRSEs
%t1H = findobj(gcf,'tag','Targ1Text');
set(t1H,'TooltipString',num2str(niirse(1)));
set(t1H,'String',num2str(niirse(1)));
set(t2H,'TooltipString',num2str(niirse(2)));
set(t2H,'String',num2str(niirse(2)));
set(t3H,'TooltipString',num2str(niirse(3)));
set(t3H,'String',num2str(niirse(3)));
set(t4H,'TooltipString',num2str(niirse(4)));
set(t4H,'String',num2str(niirse(4)));
set(t5H,'TooltipString',num2str(niirse(5)));
set(t5H,'String',num2str(niirse(5)));
% Switch NIIRSE labels on or off
if NiirseLabels_sw,
  %set(t1H,'ForegroundColor',[0 .75 0]);
  set(t1H,'ForegroundColor',[0 0 0]);
  set(t2H,'ForegroundColor',[0 0 0]);
  set(t3H,'ForegroundColor',[0 0 0]);
```

```
  set(t4H,'ForegroundColor',[0 0 0]);
  set(t5H,'ForegroundColor',[0 0 0]);
else,
  set(t1H,'ForegroundColor',[1 1 1]);
  set(t2H,'ForegroundColor',[1 1 1]);
  set(t3H,'ForegroundColor',[1 1 1]);
  set(t4H,'ForegroundColor',[1 1 1]);
  set(t5H,'ForegroundColor',[1 1 1]);
end,
% Display Route Distance and Duration
% truncate values after one decimal place
travel_distance_km = round(10.*travel_distance_km)./10;
travel_time_minutes = round(10.*travel_time_minutes)./10;
%disH = findobj(gcf,'tag','DistanceBox');
set(disH,'String',[num2str(travel_distance_km),'km']);
%durH = findobj(gcf,'tag','DurationBox');
set(durH,'String',[num2str(travel_time_minutes),'min']);
% Compute slact time
slack = TOTlimit - travel_time_minutes;
% truncate slack after one decimal place
slack = round(10.*slack)./10;
% Write to slack text box
%sH = findobj(gcf,'tag','slackbox');
if slack >= 0,
  set(sH,'BackgroundColor',[0 .5020 .2510]);
else,
  set(sH,'BackgroundColor',[1 0 0]);
end,
slackstring = [num2str(slack),'min'];
set(sH,'String',slackstring);
% Plot Route Survivability
%rsH = findobj(gcf,'Tag','Survivability');
set(rsH,'YData',[0 survivability]);
risk,
plotm(targ_gd(5,1),  targ_gd(5,2),  'g*'),
```

```
% setup 333.m
%
% initialization script for build3 thesis code
% load map variables
load usalo
% initiate figure/GUI
%open oh2.fig,
%open guimain3.fig,
% Load max allowable reroute duration due to future mission TOT
TOTlimit = 130; % [minutes]
% Set Survivability scaling factor
survive_factor = 800; %oh2
% Initialize MinNiirse
MinNiirse = 5;
NiirseLabels_sw = 1;
sensortype = 1;
% Load EO/IR sensor NIIRS performance baseline representing
%  calculated data from GIQE v.4 (fictional data)
IRdata = [[0 8.5];[10 8];[20 7.5];[30     6.5];[40 5.5];...
```

```
      [60 4.5];[80 3.5];[100 2.5];[120 1.5];[140 0]];
EOdata = [[0 9];[10 8.5];[20 8];[30 7];[40 6];...
      [60 5];[80  4];[100 3];[120 2];[140 0]];
% Sensor Image Data Collection Time Requirements [sec]
EOtime = 4.7;
IRtime = 6.1;
SARtime = nan; %SARtime = 100;
imagetime = [EOtime ; IRtime ; SARtime];
% Input recon targets
% [lat lon alt minlookangle maxlookangle priority]
% [lat lon alt] = [deg deg m]
% lookangles = [degrees]
% priority = [scalar 0:10]
town = [40.0681  -92.8566];
targ_gd = [[38.4886  -84.5273 1 0 0 1];... % targ 1
    [40.1412  -84.2855  1 0 0 1];... % targ 2
    [ 40.2028  -79.4500 1 0 0 1];... % targ 3
    [38.4255  -79.1277  1 0 0 1];... % targ 4
    [ 41.6944  -81.0619 1 0 0 1]];... % targ 5- new
targ_gr = [pi/180*targ_gd(:,1:2) targ_gd(:,3:6)];
% input threats
% 'rcs' = UAV Radar Cross Section param.
rcs = 1;
% threat altitude = 1m (ground level)
thtalt = 1;
% 'tht' = [lat lon alt [threat range radius]] = [deg deg m m]
  tht_gd = [[41.5019  -79.1277 thtalt 120*1852]];% (m/nm=1852),
111120m=60nm;
% convert deg to rads for lla2ecef.m
tht_gr = [tht_gd(:,1:2).*pi/180  tht_gd(:,3:4)];
% 'rmarr' = Range for Maximum Acceptable Radar Return [m]
rmarr = rcs*tht_gr(:,4);
% convert tht to ECEF coords for analysis
tht_e = lla2ecef(tht_gr(:,2), tht_gr(:,1), tht_gr(:,3));
tht_e(:,4) = tht_gr(:,4);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% INITIALIZATION For Main Code: %%%%%%
% aircraft setup
% map setup
maplatlimit = [ 37  43];
maplonlimit = [-78 -86];
% set selection tolerance for mouse-selecting waypoints
selecttol = 70;
% set route point resolution quality factor; (not = exact # pts)
resolution = 100;
% color for map background
mapbackgrndclr = [1 .97 .99];
% waypoints input by user
xuser = 0;
yuser = 0;
format compact,
run = [];
initial_run = 1;
%%%%%% FOR INITIAL RUN:
```

```
% for oh2.fig:
  %IR, MinNiirse= 5, labels on
  y1 = [37.5200; 38.7563; 40.1566; 41.4985; 41.3474; 41.1352; 39.9407;
40.0950; 39.9716; 38.2833; 37.1033];
  x1 = [ -85.7966; -84.6280;-83.9430; -81.6461; -81.5656; -81.9685; -
80.9208; -79.7522; -79.5911; -79.4702; -84.3057];
npath1 = length(x1);
% load initial waypoints as user input pts.
xuser = x1;
yuser = y1;
% set initial values
x = x1;
y = y1;
% z = constant = WGS-84 Elipsoidal Altitude vector [m]
alt_m = 20000,
alt_nm = alt_m/1852,
z1 = ones(npath1,1)*alt_m;
% velocity at each pt [m/s]
vel = 180; % 180m/s = 350 knots TAS
v1 = ones(npath1,1)*vel;
% Set up all Object Handles
rsH = findobj(gcf,'Tag','Survivability');
sH = findobj(gcf,'tag','slackbox');
mapH = findobj(gcf,'Tag','MapAxis');
disH = findobj(gcf,'tag','DistanceBox');
durH = findobj(gcf,'tag','DurationBox');
t1H = findobj(gcf,'tag','Targ1Text');
t2H = findobj(gcf,'tag','Targ2Text');
t3H = findobj(gcf,'tag','Targ3Text');
t4H = findobj(gcf,'tag','Targ4Text');
t5H = findobj(gcf,'tag','Targ5Text');
```

```
function pathdata = checkpath(path1,npath1,az2targ,el2targ,sensortype),
%
% This function populates matrix 'pathdata' with columns
%   of data corresponding to various a/c path states
% Each row of 'pathdata' corresponds to the same row # of path1
% Input parameters:
%   path1(path1_gd) :  (length = npath1)
%  [lat lon alt vel segment# noturn]=[deg deg m  m/s 1,2...  0/1]
%   sensortype = [1/2/3]: 1=EO 2=IR 3=SAR
%
% Output parameters:
%   pathdata :          (length = npath1)
% [noturn EOok IRok SARok]
%
% Sensor Field of Regard Parameters = [rad]
EOazlim = 15*pi/180;
EOellim = 80*pi/180;
IRazlim = 15*pi/180;
IRellim = 80*pi/180;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% initialize noturn
noturn = path1(:,6);
% initialize pathdata:
```

```
pathdata = zeros(npath1,1);
% set noturn=0 for each point immediately before a turn
% these points are duplicates created during path interpolation
noturnshift = noturn - ones(npath1,1);
noturn = noturn + [noturnshift(2:npath1);0];
% assign noturn to pathdata column
pathdata(:,1) =  noturn;
% check EOok: check acceptable a/c FOR
if sensortype == 1,
  for p = 1:(npath1-1),
  % test noturn (skip current p-index if turning)
  if noturn(p,1),
  if ((az2targ(p) >= (pi/2-EOazlim) & az2targ(p) <= (pi/2+EOazlim)) |
...
  (az2targ(p) >= (pi*3/2-EOazlim) & az2targ(p) <= (pi*3/2+EOazlim))) &
...
   el2targ(p) < EOellim,
   pathdata(p,2) = 1;
   end,
   end, % if noturn
   end,% for p
  % fill IR and SAR OK comps in pathdata
  pathdata(:,3:4) = ones(npath1,2);
end, %if sensortype
% check IRok: check acceptable a/c FOR
if sensortype == 2,
  for p = 1:(npath1-1),
  % test noturn (skip current p-index if turning)
  if noturn(p,1),
  if ((az2targ(p) >= (pi/2-IRazlim) & az2targ(p) <= (pi/2+IRazlim)) |
...
  (az2targ(p) >= (pi*3/2-IRazlim) & az2targ(p) <= (pi*3/2+IRazlim))) &
...
  el2targ(p) < IRellim,
  pathdata(p,3) = 1;
  end,
  end, % if noturn
end,% for p
% fill EO and SAR OK comps in pathdata
  pathdata(:,2) = ones(npath1,1);
  pathdata(:,4) = ones(npath1,1);
end, %if sensortype
% check SARok: check acceptable a/c FOR
%%% Not currently implemented
% store old path before proceeding
path0_gd = path1_gd;
npath0 = length(path0_gd(:,1));
% input new trajectory for comparison
% plot start and end points
plotm(path1_gd(1,1:2),'k*'),
plotm(path1_gd(npath1,1:2),'k*'),
% set n to retain first path point
n = 1;
x=[];y=[];xi=[];yi=[];
% restore original start point
```

```
x(1,1) = x1(1,1);
y(1,1) = y1(1,1);
disp('Left-Click to Mark New Trajectory.'),
disp('Right-Click to Finish.'),
run1 = 1;
btype = 'normal';
while run1 ~= 0,
  [yi,xi] = inputm(1);
  btype = get(gcf,'selectiontype');
  switch btype;
  case 'normal';
    % plot the point just entered
    plotm(yi,xi,'bo');
    n = n + 1;
    x(n,1) = xi;
    y(n,1) = yi;
    % plot updated route
    plotm(y,x,'b'),
  otherwise,
    run1 = 0;
  end
end
% restore original end points
x = [x(:,1) ; x1(length(x1),1)];
y = [y(:,1) ; y1(length(y1),1)];
% update npath1 to user input points
npath1 = length(x);
% temp store user input for plotting separately
xuser = x;
yuser = y;
% plot start and end points
plotm(path0_gd(1,1:2),'k*'),
plotm(path0_gd(npath0,1:2),'k*'),
```

```
% adjust.m
%
% input new trajectory for comparison
disp('1. Add new waypoint: Select a route segment, then add a new
waypoint to it.'),
disp('   -OR-'),
disp('2. Move a waypoint: Select a waypoint, then click to mark its new
location.'),
disp(' '),
disp('Right-click to finish'),
x =[]; y =[]; xi =[]; yi =[];
run1 = 1;
btype = 'normal';
while run1 ~= 0,
y0 = path0_gd(:,1); x0 = path0_gd(:,2);
% input new trajectory for comparison
% first click selects seg to add a waypt, or waypt to move
[yi,xi] = inputm(1);
btype = get(gcf,'selectiontype');
switch btype;
case 'normal';
```

```
% check if close to an existing waypoint (_user)
dpath00 = [xuser,yuser] - ones(length(xuser),1)*[xi yi];
dpath00 = sqrt(dpath00(:,1).^2 + dpath00(:,2).^2);
[dpath00sort,orderp0] = sortrows(dpath00);
if dpath00sort(1) <= abs((maplonlimit(2) - maplonlimit(1))/selecttol),
% second click defines new position location (or breaks if rt click)
            [yi,xi] = inputm(1);
            x = xuser;
            y = yuser;
            x(orderp0(1)) = xi;
            y(orderp0(1)) = yi;
          else,
            dpath0 = [x0,y0] - ones(length(x0),1)*[xi yi];
            dpath0 = sqrt(dpath0(:,1).^2 + dpath0(:,2).^2);
            [dpath0sort,orderp] = sortrows(dpath0);
% before here, test if close to a user pt; if not, continue below
            % if yes, just move it; don't create a new pt.
            % store seg# from selected pt.
            newptseg = path0_gd(orderp(1,1),5);
% second click defines new position location (or breaks if rt click)
            [yi,xi] = inputm(1);
            for n = 1:(length(xuser)+1),
              if n < newptseg + 1,
                x(n,1) = xuser(n);
                y(n,1) = yuser(n);
              elseif n == newptseg + 1,
                x(n,1) = xi;
                y(n,1) = yi;
              elseif n > newptseg + 1,
                x(n,1) = xuser(n-1);
                y(n,1) = yuser(n-1);
              end, %if
            end, %for n
          end, %else select waypoint test
```

```
% create.m
%
          % store previous userpath before update
          xuser0 = xuser;
          yuser0 = yuser;
          % update userpath
          xuser = x;
          yuser = y;
          npath1 = length(x);
          % update needed portions of path0_gd
          segment =[]; xfit =[]; yfit =[];
          for p = 1:(npath1-1), % pre-interpolated # of path pts.
            xtemp = []; ytemp = [];
            [ytemp, xtemp] = interpm(y(p:p+1),x(p:p+1),maxdiff);
            % build x,y columns of path1
            xfit = [xfit ; xtemp];
            yfit = [yfit ; ytemp];
            % build 5th column of path1 = [path segment number]
            segment = [segment(:) ; p*ones(length(xtemp),1)];
          end
```

```
          npath0 = length(yfit);
          % geodetic frame [lat lon [] [] seg# [] ]
          path0_gd =
[yfit,xfit,zeros(npath0,2),segment,zeros(npath0,1)];
          % plot last route colored
          plotm(yuser0,xuser0,'c--'),
          % plot updated route and waypoints
          plotm(yuser,xuser,'b.'),
          plotm(yuser,xuser,'b'),
          %n = n + 1;
          %x(n,1) = xi;
          %y(n,1) = yi;
          %btype = get(gcf,'selectiontype');
        otherwise,
          run1 = 0;
          % erase last point entered (w/ right click)
          %x = x(1:n-1,1);
          %y = y(1:n-1,1);
        end, %switch
      end, %while run1
      % plot start and end points
      plotm(path0_gd(1,1:2),'k*'),
      plotm(path0_gd(npath0,1:2),'k*'),
      % set n to retain first path point
      %n = 1;
      %x =[]; y =[]; xi =[]; yi =[];
      % restore original start point
      %x(1,1) = x1(1,1);
      %y(1,1) = y1(1,1);
      % restore original end points
      %x = [x(:,1) ; x1(length(x1),1)];
      %y = [y(:,1) ; y1(length(y1),1)];
      % update npath1 to user input points
      % temp store user input for plotting separately
      xuser = x;
      yuser = y;
```

```
function drawthreatring = ring(center,radius,color)
%
% requires: center = [lat, lon] = [degrees]
%           radius = [m]
earthradius = almanac('earth','radius','m');
[latc,lonc] = scircle1(center(1),center(2),radius,[],earthradius);
plotm(latc,lonc,color),
```

```
function drawvisibilityarc = arc(center,...
   minlook,maxlook,arcrad,arccolor);
%
% requires: center = [lat, lon] = [deg]
%           min/maxlook         = [deg]
%           arcrad              = [m]
%           arccolor            = 'x'
```

```
earthradius = almanac('earth','radius','m');
[latc,lonc] = scircle1(center(1),center(2),arcrad,[minlook
maxlook],earthradius);
plotm(latc,lonc,arccolor),
```

```
function heading = crad2heading(crad)
%
% function heading = crad2heading(crad)
%
% This function converts an angle (in radians) referenced
% from the quadrant I Cartesian x-axis to an azimuthal
% heading reference in degrees:
% North = 0   degrees
% East  = 90  degrees
% South = 180 degrees
% West  = 270 degrees
%
% Allows column vector inputs to 'crad'
% Example:
% aircraft_heading_in_degrees = crad2heading(atan2(y,x))
cdeg = crad.*(180/pi);
n = length(crad);
for i = 1:n,
   if ((cdeg(i) <= 90) & (cdeg(i) > -180)),
      heading(i,1) = 90 - cdeg(i);
   elseif ((cdeg(i) > 90) & (cdeg(i) <= 180)),
      heading(i,1) = (180 - cdeg(i)) + 270;
   else
      heading = nan,
      %heading = -500*ones(n);
   end
end
```

```
function  ECEF_pos = lla2ecef(lon, lat, alt)
%
% function  ECEF_pos = lla2ecef(lon, lat, alt)
% This function converts from geodetic coordinates (longitude,
% latitude, and altitude) to an ECEF position vector.
% Input parameters:
%   lon : WGS-84 geodetic longitude (rad)
%   lat : WGS-84 geodetic latitude (rad)
%   alt : WGS-84 ellipsoidal altitude (m)
% Output parameter:
%   ECEF_pos : ECEF position vector (m)
% initial conditions
a = 6378137;
e2 = 0.00669437999013;
n = length(lon);
rn = a./sqrt(ones(n,1)-e2.*(sin(lat)).^2);
R = (rn + alt).*cos(lat);
ECEF_pos(:,1) = R.*cos(lon);
ECEF_pos(:,2) = R.*sin(lon);
ECEF_pos(:,3) = (rn.*(1-e2) + alt).*sin(lat);
```

```
function [lon_factor, lat_factor] = rad2meter(latitude, wgs84_alt)
%
% function [lon_factor, lat_factor] = rad2meter(latitude, wgs84_alt)
% This function calculates the conversion factor to go from radians
% to meters for both longitude and latitude
% (latitude = [rad]; wgs84_alt = [m])
a=6378137.0;   % WGS-84 values
e2=0.00669437999013;
sin2lat=(sin(latitude))^2;
Rm=a*(1-e2)/((1-e2*sin2lat)^(3/2));
lat_factor=Rm + wgs84_alt;
Rp=a/sqrt(1-e2*sin2lat);
lon_factor=cos(latitude)*(Rp + wgs84_alt);
```

Bibliography

[1] Air Force Scientific Advisory Board. Technologies and Combat Operations: Executive Summary. 6 December 1996. Retrieved September 3, 1999 from the URL, http://web.fie.com/htdoc/fed/afr/sab/any/text/any/afrttech.htm.

[2] Alber, Harold. Global Hawk Future Payload Sensors (ASC/RAVS), Wright-Patterson AFB, OH. Personal Interview. 7 February 2000.

[3] Daly, Paul Major USAF. Deputy Director, ACC Global Hawk Assessment Team, Edwards AFB CA. Telephone interview. 2 November 1999.

[4] Daly, Paul Major USAF. Global Hawk Mission Planning System Program Review. San Diego, CA. 18 August 1999.

[5] Dornheim, Michael A. "Apache Tests Power Of New Cockpit Tool," Aviation Week & Space Technology, 18 October 1999: 46-49.

[6] Flood, Randy A. A JAVA Based Human Computer Interface for a UAV Decision Support Tool Using Conformal Mapping. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1999 (AD-A361562)(0117201210665).

[7] Francis, Michael S. "The Uninhabited Tactical Aircraft: Redefining the Air Combat System," Aerospace America, Vol. 36 No. 5: 44-50 (May 1998).

[8] Gentner, Frank C., Michael S. Kettel and Mona J. Crissey. Automated Aeronautical Mission Planning Systems: Implicaitions for Trainers. Army Simulation, Training, and Instrumentation Command, Orlando FL, 1998 (AD-A362204)

[9] Gerber, Garth. Global Hawk AV2-07 and AV2-08 EO/IR Incentive Assessment Results. ERIM International. Presentation to ASC/RAV, Wright-Patterson AFB, OH. August 1999.

[10] Headquarters US Atlantic Command. HAE UAV Joint Employment Concept of Operations. Norfolk: HQ USACOM, 15 July 1998.

[11] Hughes Aircraft Company. Global Hawk/Tier II+ EO/IR Imaging System Program Final Design Review. Subcontract 69-973-008D. El Segundo, CA. April 1996.

[12] Johnson, Rubin and Michael Roach. ORCA Corporation. Whittier, CA. Personal Interview. 26 October 1999.

[13] Johnson, Rubin and Michael Roach. The Inflight Mission Planning System: Final Report, September 1996-May 1999. ORCA Corporation. Contract F33615-96-C-3610. Human Effectiveness Directorate, AFRL, Wright-Patterson AFB OH, May 1999.

[14] Kaplan, Elliott D. and J. L. Leva. Understanding GPS: Principles and Applications. Boston: Artech House Publishers, 1996.

[15] National Imagery and Mapping Agency. General Image Quality Equation Version 4.0 User's Guide. Washington DC. December 1996.

[16] O'Rourke, Kevin P. Captain, USAF. Dynamic UAV Routing with a Java-encoded Reactive TABU Search Metaheuristic. Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1999 (AD-A361708)(0117201209345).

[17] OR Concepts Applied Corporation. Low Observable Inflight Mission Planning Review #5. Whittier, CA. March 1999.

[18] Pendleton, Ryan R. Captain, USAF. Chief, Intelligence Operations, 31st Fighter Wing Aviano AB, Italy. Personal Interview. 10 June 1999.

[19] Riley, Victor. "What Avionics Engineers Should Know About Pilots and Automation," Proceedings of the AIAA/IEEE 14th Digital Avionics Systems Conference. 252-257. New York: IEEE Press, 1995.

[20] Rodriguez, Julio and Doug Chapman. ASC/RAVT, Wright-Patterson AFB OH. Personal interview. 26 August 1999.

[21] Silverthorn, James. Science Applications International Corporation, Systems and Technology Division. Arlington, VA. Telephone Interview. 4 November 1999.

[22] Smerke, Robert P. and Vincent C. Anderson. "Pilot-Vehicle Integration: An Overview for Conceptual Design," World Aviation Congress. Los Angeles CA, October 21-24 1996.

[23] Stokes, Alan, Christopher Wickens, and Kirsten Kite. Display Technology: Human Factors Concepts. Warrendale, PA: society of Automotive Engineers, Inc, 1990.

[24] Vian, John L. and John R. Moore. "Trajectory Optimization with Risk Minimization for Military Aircraft," AIAA Journal of Guidance, Control, and Dynamics Vol.12 No.3: 311-317 (May-June 1989).

Vita

Captain David Edward-Robert Pritchard was born 15 December 1971 in Marquette, MI. After graduating from high school in 1990 and spending his first year of college at Northern Michigan University in Marquette, he entered the Mechanical Engineering department at Michigan Technological University, Houghton, in Upper Michigan's Keweenaw Peninsula. He received a Bachelor of Science degree Magna Cum Laude in Mechanical Engineering and his Air Force commission via ROTC from MTU on November 19th, 1994.

His first Air Force assignment was from 1995 to 1996 as Integration and Test Director for the High-Altitude Balloon Experiment (HABE) at the Air Force Research Laboratory, Kirtland AFB, New Mexico. HABE is a balloon-borne, near-space environment experiment supporting the Space Based Laser (SBL) demonstrating precision tracking and targeting of a boosting theater ballistic missile. Captain Pritchard also managed HABE's Scaled Rocket Test Program, designing, building, testing, and flying 1 m long model rockets as tracking targets for ground testing the 7000 lb HABE payload.

From 1996 to 1998 at Kirtland AFB, Capt Pritchard led the Skunkworks Mission 9 (SM-9) design team for the Countermeasures Hands-On Program (CHOP). CHOP supports the development of robust theater and national missile defenses by investigating potential threats and countermeasures from the perspective of likely adversarial nations. The tri-service SM-9 team of 6 military and civil service engineers from the US and

Great Britain designed, built, and flight tested a new missile countermeasure system intended to defeat US ballistic missile defensive systems. Their work encompassed the countermeasure system's entire life cycle, producing a viable test article within one year.

He was then selected to attend the Air Force Institute of Technology School of Engineering to receive a Masters of Science Degree in Aeronautical Engineering. His follow-on assignment after AFIT is to the US Strategic Command's Force Assessments/ Penetration Analysis branch (USSTRATCOM/J534) at Offutt AFB in Omaha, Nebraska. Captain Pritchard is married to Heidi L. Pritchard.

Permanent Address:
555 Vallombrosa #9
Chico, CA 95926

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 2000

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
DYNAMIC ROUTE REPLANNING AND RETASKING OF UNMANNED AERIAL RECONNAISSANCE VEHICLES

5. FUNDING NUMBERS

6. AUTHOR(S)
David E. R. Pritchard, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GAE/ENY/00M-10

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
ASC/RAV
Attn: Mr. Vernon Best II
2460 Loop Road West
WPAFB OH 45433-7106

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Prof. C. H. Spenny, ENY, DSN: 785-6565, ext. 4320

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Many common operational situations in High Altitude Endurance (HAE) UAV reconnaissance necessitate more direct human involvement in the aircraft control process than is currently acknowledged or allowed. The In-Flight Replanning System (IFRS) is developed as a notional solution, with the Global Hawk HAE UAV as the reconnaissance system model. Problematic scenarios for current in-flight mission replanning systems are presented and revisited with the IFRS, demonstrating improved replanning results.

14. SUBJECT TERMS
Dynamic Replanning, Dynamic Retasking, Global Hawk, Mission Planning, Reconnaissance, UAV

15. NUMBER OF PAGES
87

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL